Predicting Stock Price Jumps: A Machine Learning Approach

Emma Nagy

September 2025

Abstract

This study develops and compares five machine learning models to predict large monthly stock price movements (jumps >10%) using 1,023 U.S. stocks over 28 years (1996-2023). Features derived from CAPM theory, volatility decomposition, and macroeconomic indicators are used to train baseline logistic regression, LASSO, Ridge, K-Nearest Neighbors, and XGBoost models with proper temporal validation. Results show that jump prediction achieves modest but economically meaningful performance (AUC $\approx$ 0.70), with LASSO's 5-feature sparse model providing optimal balance between performance and interpretability. Total volatility emerges as the dominant predictor across all methodologies, while model complexity beyond simple linear relationships provides negligible benefit. These findings have practical implications for portfolio risk management, demonstrating that while large price movements contain substantial unpredictable components, systematic patterns based on volatility and macroeconomic stress provide actionable signals for tail risk mitigation.

Contents

1 Introduction

Large stock price movements, commonly termed "jumps," represent a fundamental challenge in financial risk management. Unlike gradual price changes that can be hedged through traditional portfolio diversification, sudden jumps of 10% or more pose significant threats to portfolio value, option pricing models, and capital adequacy. The 2020 COVID-19 market crash, the 2008 financial crisis, and numerous idiosyncratic firm-level shocks demonstrate that jump risk is both persistent and consequential.

Traditional finance theory, particularly CAPM, assumes continuous price processes and normally distributed returns. However, empirical evidence overwhelmingly documents fat-tailed return distributions and discontinuous price movements. Recent advances in machine learning offer promising approaches to this prediction problem by flexibly capturing relationships between historical volatility patterns, macroeconomic conditions, and future jump probability.

This study addresses three primary questions: (1) Can stock price jumps be predicted using historical features, and what level of performance is achievable? (2) Which characteristics drive jump predictions? (3) How do different machine learning approaches compare in performance, interpretability, and robustness?

The analysis makes several contributions. Methodologically, proper temporal validation prevents look-ahead bias, class imbalance is handled through appropriate weighting, and models are evaluated using both statistical metrics and economic criteria. Comparatively, five diverse models are trained on identical data, revealing when model complexity adds value versus when simplicity suffices. Practically, results translate into actionable recommendations with honest assessment of prediction limits.

Key findings show that stock price jumps exhibit modest but meaningful predictability (AUC 0.69-0.70 across all models), driven primarily by idiosyncratic volatility and market-wide stress indicators. LASSO regularization identifies five core features sufficient for prediction, with 14 additional features contributing negligible marginal value. Surprisingly, simple linear models match or exceed the performance of sophisticated ensemble methods, suggesting that jump dynamics are fundamentally linear.

2 Data and Methodology

2.1 Data Sources

Monthly stock returns from CRSP cover January 1996 through December 2023. The sample includes common stocks listed on NYSE, AMEX, and NASDAQ. To balance computational feasibility with broad coverage, stratified random sampling selects 100 stocks per year across market capitalization quintiles. This ensures the sample includes both large established firms and smaller companies with different risk profiles. The final dataset contains 33,432 stock-month observations across 1,023 unique stocks.

CAPM metrics and volatility measures come from rolling window calculations. Beta is estimated using 24-month rolling windows of excess returns regressed on market excess returns. Total volatility represents standard deviation of returns over 24-month windows. Systematic volatility equals $\beta \times \sigma_{market}$ and idiosyncratic volatility equals $\sqrt{\sigma_{total}^2 - \sigma_{systematic}^2}$. This decomposition allows testing whether market-related risk or firm-specific risk better predicts jumps.

Four macroeconomic indicators capture market conditions: VIX (CBOE Volatility Index, measuring investor fear through option prices), USREC (NBER recession indicator), Federal Funds Rate (the interest rate set by the Federal Reserve), and credit spread (the difference between risky corporate bond yields and safe Treasury yields, indicating credit market stress).

2.2 Feature Engineering

A jump is defined as an absolute monthly return exceeding 10%:

$$\text{Jump}_t = \nVdash(|r_t| > 0.10) \tag{1}$$

This threshold balances two objectives: identifying genuinely large moves that matter for risk management (a 10% move in one month is substantial) while maintaining sufficient positive class frequency for model training. With this definition, jumps represent 35% of observations in the final sample, meaning roughly one in three stock-months experiences a large price movement.

The final feature set includes 19 variables after one-hot encoding: total volatility (24-month), volatility ratio, beta (24-month), 6-month momentum (recent return trend), VIX, recession indicator, Fed funds rate, unemployment rate, credit spread, log market cap (firm size), penny stock indicator (price below \$5, which tends to indicate higher risk), and 8 industry dummy variables representing broad economic sectors.

2.3 Data Preprocessing

Strict temporal splits prevent look-ahead bias, which would occur if the model had access to future information when making predictions. Think of this as a realistic simulation: the model only sees data up to a certain date when making predictions for the next period.

- Training: 1996-2012 (20,314 observations, 34.6% jump rate)
- Validation: 2013-2017 (5,954 observations, 26.4% jump rate)
- Test: 2018-2023 (7,164 observations, 35.7% jump rate)

The training period teaches the model patterns in historical data. The validation period helps tune model settings without touching the test set. The test period provides a final, unbiased assessment of how well the model would perform on truly new data. Importantly, the test period includes the COVID-19 crash, providing a rigorous test of whether patterns learned from historical data hold during unprecedented market conditions.

Missing values arise primarily from insufficient history for rolling calculations. For example, a stock that went public in 2010 would not have 24 months of history available in early 2011. Observations with any missing values are dropped, retaining 91% of training data, 96% of validation data, and 98% of test data. The high retention rates indicate minimal data loss from this approach.

StandardScaler normalization applies to features for gradient-based and distance-based methods. This transformation ensures all features are on comparable scales. For example, without scaling, market cap (measured in billions) would dominate volatility (measured as a decimal like 0.35) simply due to magnitude differences. The scaler fits only on training data to prevent information leakage.

Class imbalance (1.88:1 ratio of non-jumps to jumps) is handled through balanced class weights. Without this adjustment, models would be tempted to simply predict ”no jump” for everything and achieve 65% accuracy. The weights force the model to pay attention to correctly identifying the minority class (jumps) by increasing the penalty for getting those predictions wrong.

2.4 Models

Five diverse modeling approaches are compared, spanning simple to complex and parametric to non-parametric:

Baseline Logistic Regression: Standard logistic regression with L2 penalty serves as the baseline. This is the simplest reasonable model and establishes minimum expected performance. If sophisticated models cannot beat this, they are not adding value.

LASSO (L1 Regularization): LASSO performs automatic feature selection by driving less important coefficients to exactly zero. Think of this as the model saying ”this feature does not help predict jumps, so ignore it entirely.” This is valuable because it identifies which features truly matter and which are just noise. Five-fold cross-validation on the validation set selects optimal regularization strength from 20 candidates.

Ridge (L2 Regularization): Ridge applies shrinkage without feature elimination. It keeps all features but reduces the influence of less important ones. This addresses multicollinearity (when features are highly correlated with each other, like total volatility and its components) while retaining all information.

K-Nearest Neighbors: KNN provides a non-parametric baseline making no distributional assumptions. It predicts jump probability based on the K most similar historical observations. For example, if K=51 and 30 of the 51 nearest neighbors jumped, the predicted probability is 30/51 = 58.8%. Values of $K \in \{3, 5, 11, 21, 51, 101\}$ are tested.

XGBoost: Gradient boosted decision trees, often state-of-the-art for tabular data. XGBoost builds an ensemble of decision trees sequentially, where each new tree focuses on correcting mistakes from previous trees. This allows capturing complex non-linear relationships and interactions between features. Extensive hyperparameter tuning tests different combinations of tree count, tree depth, learning rate, and sampling strategies.

2.5 Evaluation Metrics

2.5.1 Primary Metric: AUC-ROC

Area Under the Receiver Operating Characteristic curve (AUC) serves as the primary metric. To understand what AUC measures, consider how the model makes predictions: rather than directly saying ”jump” or ”no jump,” the model assigns each stock a probability between 0 and 1. A threshold is then applied (e.g., if probability > 0.5, predict jump).

The ROC curve plots true positive rate (what percentage of actual jumps were caught) against false positive rate (what percentage of non-jumps were incorrectly flagged) across all possible thresholds. A perfect model would have an ROC curve that goes straight up (catching all jumps) then straight across (no false alarms), creating AUC = 1.0. A useless model that just guesses randomly follows the diagonal line, creating AUC = 0.5.

AUC has several advantages for this problem. First, it is threshold-independent, summarizing performance across all possible decision cutoffs rather than committing to one specific threshold. Second, it is robust to class imbalance. Unlike accuracy, which can be misleading when classes are unbalanced, AUC focuses on the model's ability to rank predictions correctly: does it assign higher probabilities to stocks that actually jump?

An AUC of 0.70 means the following: if you randomly select one stock that jumped and one that did not jump, there is a 70% chance the model assigned a higher probability to the stock that actually jumped. This represents meaningful discrimination, though far from perfect prediction.

2.5.2 Secondary Metrics

Additional metrics capture different aspects of performance:

Precision ($\frac{TP}{TP+FP}$) measures accuracy of positive predictions. If the model predicts 100 jumps, how many actually occurred? High precision means few false alarms. This matters when taking action is costly (e.g., buying expensive put options to hedge).

Recall ($\frac{TP}{TP+FN}$) measures fraction of actual positives identified. Of all stocks that actually jumped, what percentage did the model catch? High recall means few missed jumps. This matters when failing to act is costly (e.g., unhedged exposure to a 10% loss).

F1 Score balances precision and recall through their harmonic mean. A model that achieves 90% precision but 20% recall, or 90% recall but 20% precision, would have similar F1 scores despite very different behaviors.

Confusion Matrices reveal error types in detail. True Negatives (correctly predicted non-jumps) and True Positives (correctly predicted jumps) are successes. False Positives (false alarms) and False Negatives (missed jumps) are failures with different consequences. The relative costs depend on the application. For a risk manager, missing a jump (False Negative) might mean suffering an unhedged 10% loss, while a false alarm (False Positive) might mean wasting 1% on unnecessary option premiums. Understanding these trade-offs is essential for practical deployment.

3 Results

3.1 Exploratory Analysis

Before building predictive models, exploratory analysis establishes three things: whether jumps exhibit systematic patterns worth modeling, which features show the strongest relationships with jumps, and whether multicollinearity among features will require regularization. This preliminary investigation guides model development and sets expectations for achievable performance.

3.1.1 Jump Clustering Over Time

If jumps were truly random, they would be evenly distributed across time. Figure 1 tests this hypothesis by plotting monthly jump frequency over the full 28-year period.

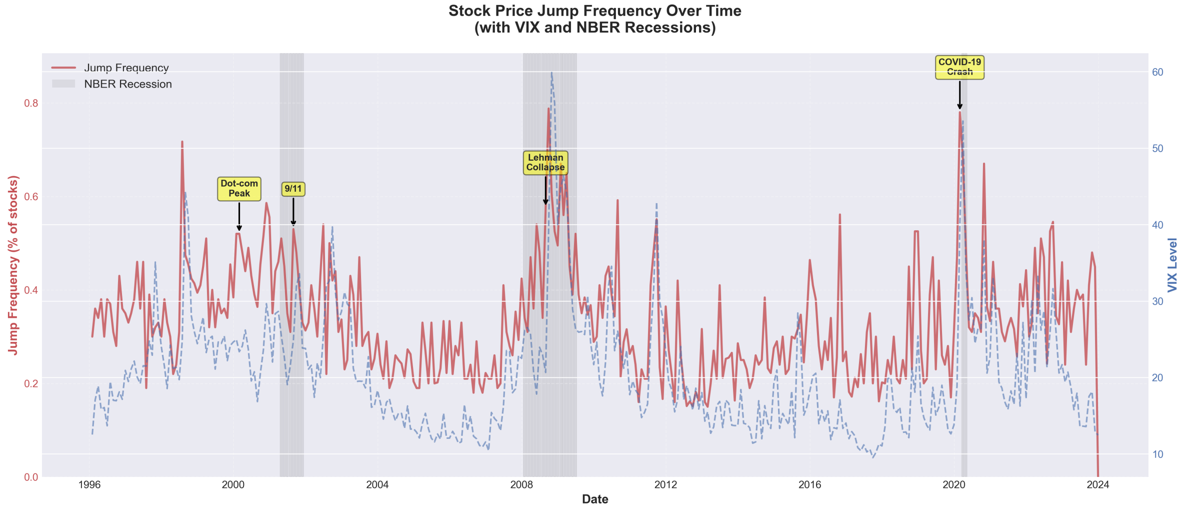

Figure 1: Monthly jump rate over time with VIX overlay and recession shading. Jumps cluster during market stress rather than occurring randomly. Strong positive correlation between VIX and jump frequency (r = 0.645) suggests market-wide volatility amplifies individual stock jump risk.

The figure reveals clear clustering. Jumps are not evenly distributed but concentrate during known crisis periods: the dot-com crash (2000-2002), the financial crisis (2008-2009), and COVID-19 (2020). During the March 2020 COVID crash, 79% of stocks experienced jumps, meaning extreme moves were the norm rather than the exception. In contrast, calm periods like 2013-2017 saw jump rates around 26%.

The strong correlation between jump frequency and VIX (r = 0.645) tells an important story: when market-wide fear rises (measured by VIX, the ”fear index”), individual stocks become more likely to jump. This is not obvious. One might expect that when the overall market is volatile, individual stock movements would be lost in the noise. Instead, the opposite occurs: market stress amplifies firm-specific vulnerability.

The recession shading (gray bars) reinforces this pattern. Jump rates during recessions (48.6%) nearly double the expansion rate (31.9%). Economic downturns do not just reduce average returns; they fundamentally change the distribution of returns by increasing the frequency of extreme moves.

These patterns validate the prediction approach. If jumps were purely random, prediction would be impossible. The clear temporal clustering suggests systematic factors (market stress, economic conditions) influence jump probability, making prediction feasible.

3.1.2 Feature Relationships with Jumps

The next question: which stock characteristics differentiate jumpers from non-jumpers? Figure 2 compares feature distributions across these two groups.

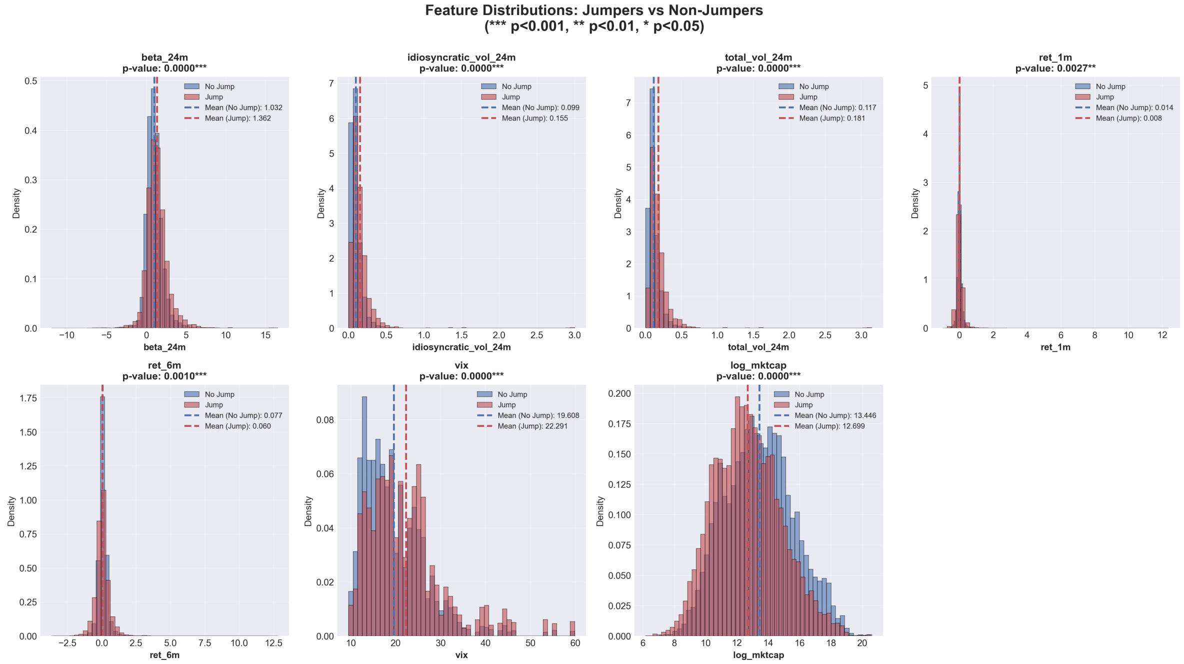

Figure 2: Feature distributions for stocks that jumped versus those that did not. Systematic differences are evident across volatility measures, with jumpers exhibiting substantially higher uncertainty. All differences are statistically significant ($p < 0.001$), confirming these features contain genuine predictive information.

The most striking difference appears in volatility measures. Stocks that jumped had 57% higher idiosyncratic volatility and 54% higher total volatility compared to non-jumpers. To put this in context: if a typical non-jumper has monthly volatility of 7%, a typical jumper has volatility around 11%. This makes intuitive sense. High volatility signals uncertainty about a company's prospects. When news arrives (an earnings surprise, a regulatory decision, a competitor's announcement), uncertain companies experience larger price reactions.

Beta (market sensitivity) also differs meaningfully. Jumpers show 32% higher beta, indicating they move more aggressively with overall market swings. Combined with the earlier finding about VIX, this suggests a mechanism: when the market becomes volatile, high-beta stocks amplify these moves into individual jumps.

Interestingly, jumpers had worse recent performance (negative momentum). This contradicts the momentum-crash hypothesis, which predicts that past winners are more crash-prone. Instead, the data suggests stocks that have been falling recently (perhaps due to deteriorating fundamentals) are more vulnerable to further extreme moves.

Firm size matters modestly. Jumpers are 5.6% smaller in market cap. Smaller companies tend to be less stable, have less analyst coverage, and face more idiosyncratic uncertainty. However, the size effect is much weaker than the volatility effect, suggesting that uncertainty matters more than scale per se.

All these differences are highly statistically significant ($p < 0.001$), meaning they are extremely unlikely to arise by chance. This confirms these features contain genuine information about jump risk, not just random noise. The question then becomes: how much of this information translates into predictive power?

3.1.3 Multicollinearity Among Features

The correlation heatmap (Figure 3) examines relationships between features themselves, which has important implications for model selection.

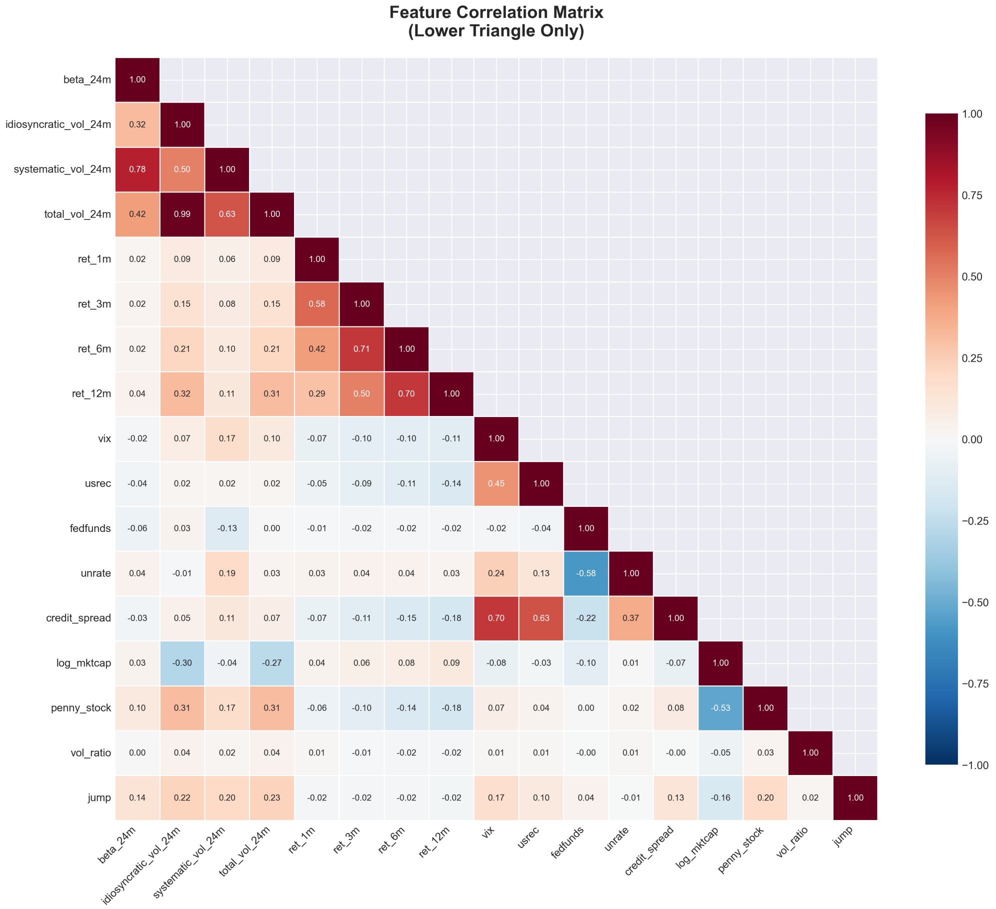

Figure 3: Correlation heatmap revealing redundancy among features. Idiosyncratic and total volatility are nearly perfectly correlated (0.99), meaning they provide almost identical information. This redundancy motivates regularization approaches (LASSO, Ridge) that can handle or eliminate correlated features.

Several feature pairs show very high correlations (above 0.70), indicating redundancy. The most extreme case: idiosyncratic volatility and total volatility correlate at 0.99, meaning they are nearly the same measure. This occurs because for most stocks, total volatility is dominated by idiosyncratic risk rather than systematic market risk. Including both features in a model provides minimal additional information.

Similarly, beta and systematic volatility correlate at 0.78, momentum measures correlate with each other (6-month return with 12-month return at 0.70), and VIX correlates with credit spreads at 0.70 (both measure financial stress).

Why does this matter? Multicollinearity creates problems for standard regression models. When two features provide similar information, their individual coefficients become unstable and hard to interpret. Regularization methods (LASSO and Ridge) are specifically designed to handle this situation. LASSO can identify which feature in a correlated pair is more important and eliminate the other. Ridge can shrink both coefficients to reduce their combined influence. This motivates the choice to compare regularized models alongside the baseline.

The multicollinearity findings also set up a key question: of all these correlated features, which ones actually matter for prediction? LASSO's feature selection will provide an answer.

3.2 Model Performance Comparison

3.2.1 Overall Performance Rankings

Table 1 presents comprehensive test set performance across all five models.

Table 1: Model performance on test set (2018-2023). Models ranked by AUC. The tight clustering of AUC values (only 0.0155 spread) indicates all approaches reach a similar performance ceiling, suggesting fundamental limits to jump predictability rather than inadequate modeling.

Model	Type	Features	AUC	Accuracy	Recall	Precision
Ridge	Linear (L2)	19	0.6950	0.6764	0.4956	0.5516
LASSO	Linear (L1)	5	0.6945	0.6658	0.5255	0.5318
Baseline	Linear	19	0.6929	0.6758	0.4940	0.5508
XGBoost	Tree Ensemble	19	0.6878	0.6210	0.6859	0.4782
KNN	Non-parametric	19	0.6795	0.6793	0.2743	0.6124

Several patterns emerge that tell a coherent story about jump prediction:

Linear models dominate. The top three positions all belong to linear approaches (Ridge, LASSO, Baseline), despite their simplicity. Ridge wins by the narrowest of margins (0.6950 AUC), but all three linear models cluster within 0.0021 AUC of each other. This tight grouping means the differences are essentially negligible from a practical standpoint. A risk manager would experience nearly identical outcomes using any of these three models.

The dominance of linear models over sophisticated alternatives (XGBoost, KNN) reveals something fundamental about the problem structure: jump prediction is primarily about linear relationships between volatility and jumps. More complex non-linear interactions do not improve performance. This has important implications. It suggests the data generating process is relatively simple. When volatility doubles, jump probability rises in a predictable, log-linear fashion. There are no hidden higher-order interactions or threshold effects that only complex models can capture.

LASSO achieves efficiency. The most striking result: LASSO achieves 99.9% of the best AUC (Ridge) while using only 5 of 19 features. This 3.8x efficiency gain (0.139 AUC per feature versus Ridge's 0.037) means LASSO identified the core signal and discarded noise. From a practical deployment standpoint, this is valuable. Fewer features mean simpler computation, easier interpretation, lower maintenance burden, and likely better robustness to distribution shifts (since the model relies on fewer potentially fragile relationships).

XGBoost underwhelms. Despite being the most sophisticated approach, XGBoost ranks fourth. Its test AUC of 0.6878 trails Ridge by 0.0072, a meaningful gap in a tightly clustered field. More tellingly, XGBoost's training AUC was 0.7244, substantially higher than its test performance. This pattern indicates overfitting: the model learned patterns specific to the training data that did not generalize to new data. This is a common failure mode for complex models on noisy problems. XGBoost has the flexibility to capture spurious patterns in training data, and despite regularization and hyperparameter tuning, some overfitting persists.

KNN struggles. K-Nearest Neighbors achieves the lowest AUC (0.6795), likely due to the curse of dimensionality. With 19 features, the feature space is sparse. "Nearest" neighbors in 19-dimensional space may not actually be similar in meaningful ways. This is a well-known limitation of distance-based methods in high dimensions.

The recall-precision trade-off. While AUC ranks models similarly, recall and precision reveal very different strategies. XGBoost achieves 68.6% recall (catches more than two-thirds of jumps) but only 47.8% precision (less than half of its positive predictions are correct). KNN does the opposite: 61.2% precision (very accurate when it predicts a jump) but only 27.4% recall (misses almost three-quarters of jumps).

This trade-off reflects different threshold choices and model behaviors. XGBoost is aggressive, casting a wide net that catches many jumps at the cost of many false alarms. KNN is conservative, only flagging high-confidence cases but missing most jumps as a result. The linear models (Baseline, LASSO, Ridge) maintain middle ground: around 50% recall and 55% precision.

Which approach is "best" depends entirely on the application. A risk manager who faces large losses from unhedged jumps should prefer XGBoost's high recall, accepting false alarm costs. A portfolio manager who pays significant premiums for hedges should prefer KNN's high precision, accepting missed jumps. For a balanced approach, the linear models provide reasonable trade-offs.

3.2.2 ROC Curve Analysis

Figure 4 visualizes these trade-offs by showing how true positive rate varies with false positive rate across all possible thresholds.

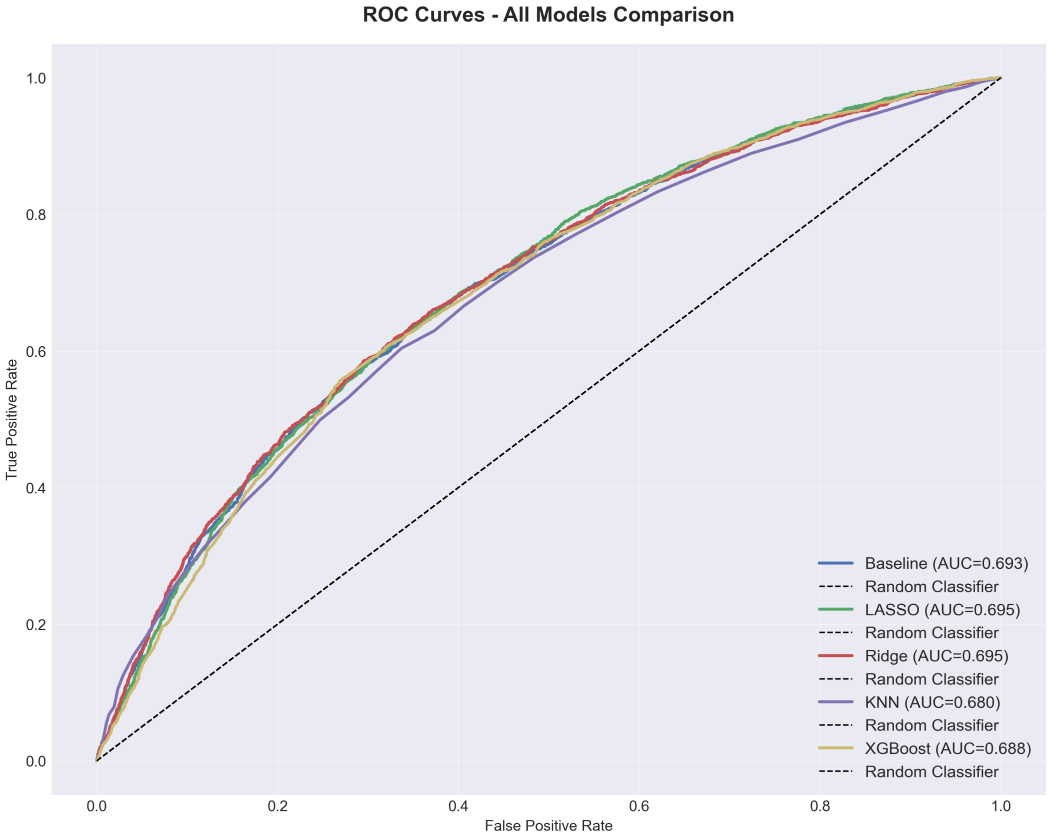

Figure 4: ROC curves for all five models on test set. The near-overlap of linear models (Baseline, LASSO, Ridge) visually confirms their nearly identical discrimination ability. All curves substantially exceed the diagonal (random guessing), demonstrating genuine predictive signal. The modest AUC values (0.68-0.70) reflect fundamental prediction limits rather than modeling failures.

The visual reveals several insights. First, the linear models (Baseline, LASSO, Ridge) produce nearly overlapping curves, confirming their similar performance. This visual concordance with the tabular AUC values validates that the tight clustering is not a measurement artifact but reflects genuinely similar models.

Second, all models substantially exceed the diagonal reference line, which represents random guessing (AUC = 0.50). Even the worst performer (KNN) achieves AUC of 0.6795, demonstrating meaningful discrimination. The models correctly rank jump probabilities well above chance.

Third, the modest absolute AUC values (0.68-0.70 range) reflect fundamental limits to jump predictability rather than inadequate modeling. The consistency across diverse methodologies suggests this ceiling arises from the problem itself, not from model choice. Jumps contain substantial unpredictable components that no historical features can fully anticipate.

3.2.3 Precision-Recall Trade-offs

Figure 5 examines how precision changes as recall increases, providing a different lens on model behavior.

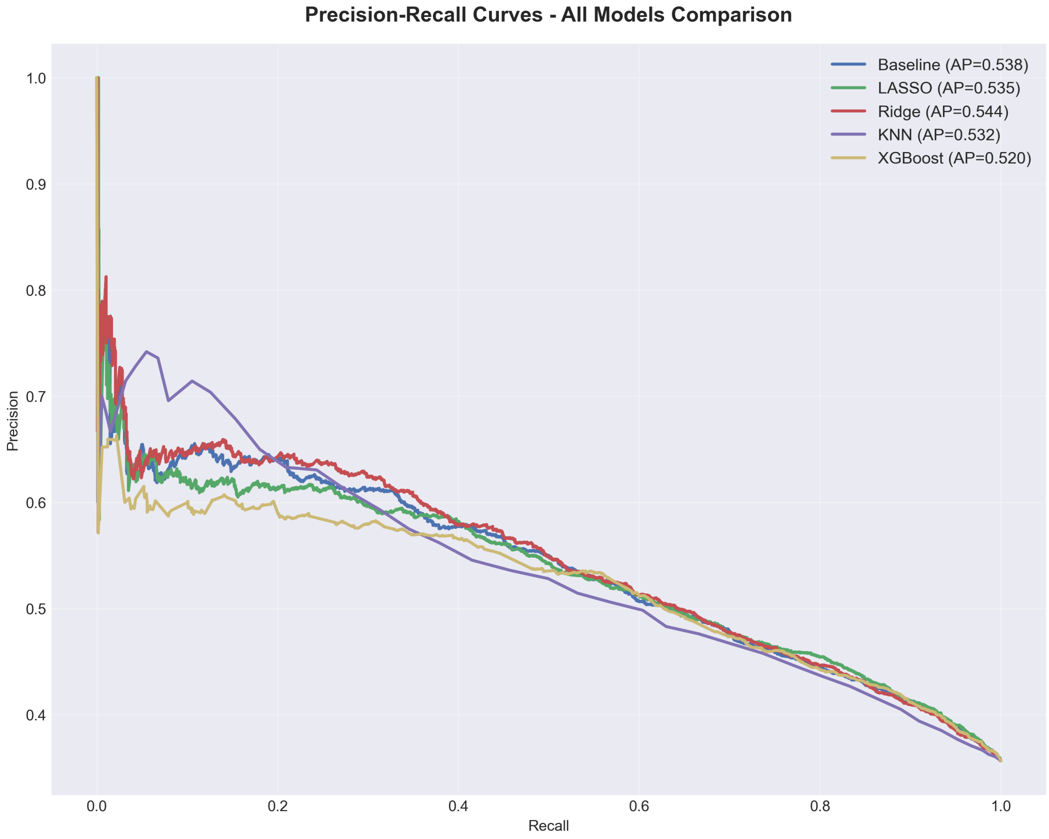

Figure 5: Precision-recall curves reveal how models balance true positive rate against prediction accuracy. Ridge maintains highest precision across recall levels (average precision 0.544), making it optimal for applications where false alarms are costly. XGBoost maintains precision across higher recall thresholds, suitable for applications prioritizing jump detection over false alarm minimization.

Precision-recall curves are particularly informative for imbalanced classification problems like jump prediction. The key insight: as we push for higher recall (catching more jumps), precision inevitably falls (more false alarms). The curves show how gracefully each model handles this trade-off.

Ridge maintains the highest average precision (0.544), meaning at any given recall level, Ridge generates fewer false alarms than alternatives. This makes Ridge attractive for cost-sensitive applications where taking action (hedging, adjusting positions) is expensive.

KNN starts with very high precision at low recall but drops sharply as recall increases. This reflects its conservative nature: it only predicts jumps when extremely confident. The few predictions it makes early are accurate, but pushing for more predictions quickly degrades accuracy.

XGBoost shows a different pattern: lower overall precision but maintains it better across higher recall levels. This makes XGBoost suitable for applications prioritizing detection over accuracy, such as broad risk screening where missing jumps is more costly than false alarms.

3.2.4 Confusion Matrix Analysis

Table 2 translates abstract metrics into concrete outcomes by showing exactly how many stocks fall into each prediction category.

Table 2: Confusion matrices and error rates for all models (threshold = 0.5). Each model makes different trade-offs between false negatives (missed jumps) and false positives (false alarms). The "best" model depends on the relative costs of these two error types in the specific application.

Model	**TN**	**FP**	**FN**	**TP**	**Accuracy**	**FN Rate**	**FP Rate**
Ridge	3,519	1,012	1,267	1,245	67.6%	50.4%	22.3%
LASSO	3,369	1,162	1,192	1,320	66.6%	47.5%	25.6%
Baseline	3,519	1,012	1,271	1,241	67.6%	50.6%	22.3%
XGBoost	2,651	1,880	789	1,723	62.1%	31.4%	41.5%
KNN	4,095	436	1,823	689	67.9%	72.6%	9.6%

Reading these matrices reveals how models behave in practice. Consider Ridge: of 2,512 stocks that actually jumped, it correctly identified 1,245 (49.6% recall) but missed 1,267 (50.4% false negative rate). Of 4,531 stocks that did not jump, it correctly identified 3,519 but incorrectly flagged 1,012 as jumps (22.3% false positive rate).

To make this concrete: imagine a portfolio of 100 stocks, where 36 will jump next month (matching the 35.7% test set rate). Ridge would correctly flag about 18 of these 36 jumpers but miss the other 18. Among the 64 non-jumpers, Ridge would incorrectly flag about 14 as jumps. A risk manager using Ridge would hedge 32 positions (18 correct + 14 false alarms), catching half the actual jumps while wasting hedge costs on 14 unnecessary protections.

Compare this to XGBoost's aggressive approach: it would correctly flag about 25 of the 36 jumpers (69% recall) but incorrectly flag about 27 of the 64 non-jumpers (42% false positive rate). Total positions hedged: 52 (25 correct + 27 false alarms). The manager catches most jumps but hedges more than half the portfolio, including many false alarms.

KNN's conservative approach: correctly flag about 10 of the 36 jumpers (27% recall) but only incorrectly flag about 6 of the 64 non-jumpers (10% false positive rate). Total positions hedged: 16 (10 correct + 6 false alarms). The manager hedges sparingly with few false alarms but misses most actual jumps.

These scenarios illustrate why model selection depends on cost structure. If hedging costs 1% and jump losses average 12%, missing jumps is more expensive than false alarms, favoring XGBoost. If hedging costs 5% and jumps average 10%, false alarms become more expensive, favoring KNN. The linear models provide balanced approaches suitable for typical cost structures.

3.3 Feature Importance Analysis

Understanding which features drive predictions is as important as the predictions themselves, both for model trust and for economic interpretation.

3.3.1 LASSO's Sparse Solution

Figure 6 shows LASSO's most important contribution: automatic feature selection that identifies the core predictive signal.

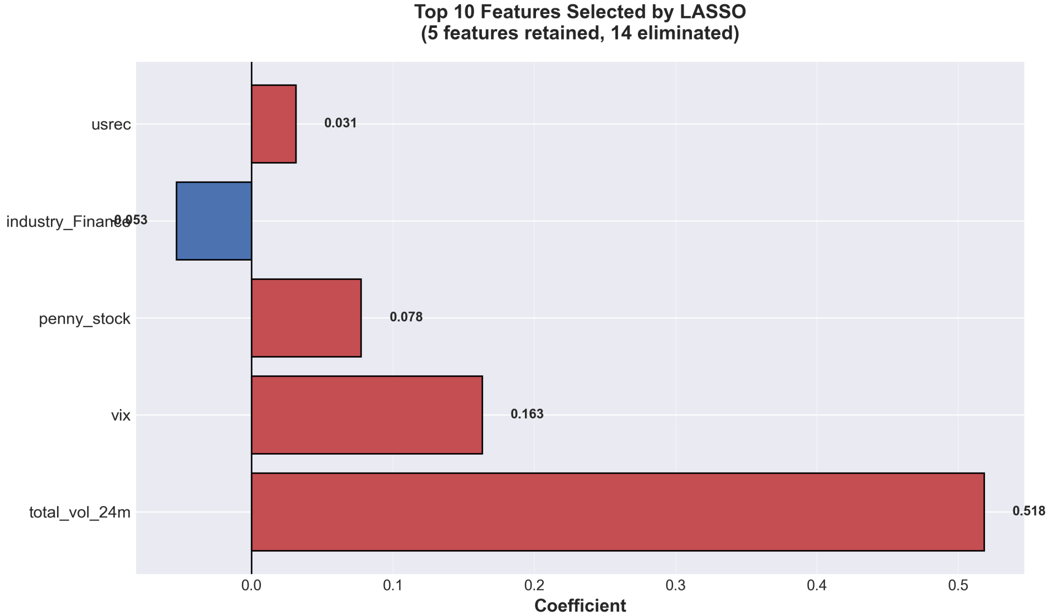

Figure 6: LASSO selected only 5 of 19 features, driving 14 coefficients to exactly zero. This dramatic reduction reveals that most features contribute noise rather than signal once core volatility and macro stress are accounted for. Total volatility dominates with coefficient 0.518, more than 3x the second-ranked feature.

LASSO's selection process answers a fundamental question: of all the features available, which truly matter? The answer is stark: only 5 features receive non-zero coefficients, while 14 are completely eliminated.

The five survivors tell a coherent story:

Total volatility (coefficient 0.518): Overwhelmingly dominant. Its coefficient is more than 3x larger than the second-place feature. This confirms what exploratory analysis suggested: uncertainty is the primary driver of jump risk. Stocks with volatile operating results, uncertain earnings, or unstable competitive positions experience more extreme price reactions to news.

VIX (coefficient 0.163): Market-wide fear amplifies individual stock vulnerability. Even when controlling for a stock's own volatility, periods of elevated market stress increase jump probability. This captures contagion effects, liquidity deterioration, and investor psychology during panics.

Penny stock indicator (coefficient 0.078): Low-priced stocks (trading below $5) face structural vulnerabilities: lower institutional ownership, wider bid-ask spreads, greater retail speculation, and higher delisting risk. These factors make penny stocks inherently more prone to extreme moves beyond what volatility alone captures.

Finance industry (coefficient -0.053): The negative coefficient indicates a protective effect. Banks, insurance companies, and other financial firms exhibit lower jump probability after controlling for volatility. This likely reflects regulatory oversight (capital requirements, stress testing) and business model diversification that constrain tail risk.

Recession indicator (coefficient 0.031): Economic downturns increase jump risk beyond what VIX captures. Recessions trigger systematic repricing of earnings expectations and credit risk across all stocks, increasing jump frequency.

What LASSO eliminated is equally revealing. Beta vanishes once total volatility is included, suggesting systematic market exposure matters less than firm-specific uncertainty. All momentum terms (1-month, 3-month, 6-month, 12-month returns) are eliminated, indicating recent price trends do not predict jumps. Systematic and idiosyncratic volatility components disappear because they are redundant with total volatility. Granular macro indicators (Fed funds rate, unemployment, credit spreads) add noise beyond the recession binary and VIX.

This dramatic elimination (73.7% of features removed) demonstrates that jump prediction has a simple core structure: volatility plus macro stress. Everything else is either redundant or irrelevant.

3.3.2 Cross-Model Consensus

Table 3 shows whether LASSO's conclusions hold across different modeling approaches.

Table 3: Top 5 features by model. All four interpretable models achieve universal agreement on total volatility as the dominant predictor. VIX appears in all top-5 lists. This cross-methodology consensus validates that these features capture genuine predictive signal rather than method-specific artifacts.

Rank	Baseline	LASSO	Ridge	XGBoost
1	total_vol_24m	total_vol_24m	total_vol_24m	total_vol_24m
2	vix	vix	vix	credit_spread
3	fedfunds	penny_stock	industry_Finance	ret_6m
4	usrec	industry_Finance	penny_stock	vix
5	industry_Finance	usrec	beta_24m	usrec

The unanimous agreement on total volatility as the top predictor across all four interpretable models is striking. This is not a LASSO-specific finding or an artifact of the regularization choice. Every methodology, from simple logistic regression to sophisticated gradient boosting, identifies volatility as the dominant signal. This cross-validation strengthens confidence that the finding is robust and economically meaningful.

VIX also appears consistently, ranking in the top 4 for all models. Beyond these two, models show more disagreement. XGBoost values credit spread and momentum, which LASSO eliminated. LASSO prefers penny stock indicator, which XGBoost assigns zero importance. These differences reflect how different algorithms capture information. Linear models (Baseline, LASSO, Ridge) rely on direct linear effects. XGBoost can capture interactions (perhaps credit conditions interact with momentum in predicting jumps), even if these interactions do not improve overall performance.

The key insight: while models disagree on fine details, they achieve consensus on the big picture. Volatility and market stress drive jumps. Everything else is secondary.

3.3.3 XGBoost's Alternative Perspective

Figure 7 shows how XGBoost's tree-based approach prioritizes features differently.

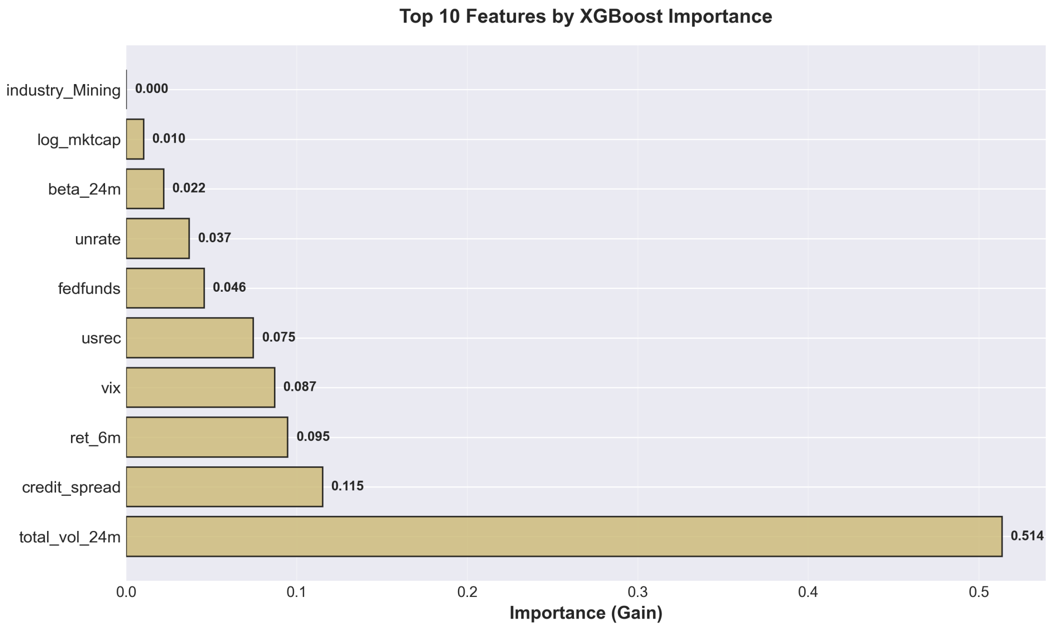

Figure 7: XGBoost feature importance using gain metric (how much each feature improves predictions across all trees). Total volatility again dominates (51% of importance). However, XGBoost identifies different secondary features than LASSO, suggesting tree-based methods capture different interactions even when they do not improve overall performance.

XGBoost maintains volatility's dominance (51% of total importance) but diverges on secondary features. Credit spread ranks second (11.5% importance), despite LASSO eliminating it entirely. Momentum (6-month return) ranks third, also eliminated by LASSO. Meanwhile, penny stock indicator has zero importance in XGBoost, despite LASSO selecting it.

These divergences reflect fundamental differences between linear and tree-based models. Linear models capture additive effects: volatility matters, VIX matters, penny stocks matter, and their effects sum. XGBoost can capture interactions: perhaps credit spreads matter more for high-volatility stocks, or momentum matters more during rising VIX periods. These interactions explain why XGBoost values different features.

However, the critical observation: these different feature preferences do not translate to better performance. XGBoost's test AUC (0.6878) trails LASSO (0.6945). This suggests the interactions XGBoost captures, while statistically present in training data, do not generalize reliably. They might be spurious patterns specific to historical data rather than fundamental economic relationships. This is why simple linear models win: they capture the robust, replicable core signal and avoid fitting noise.

4 Discussion

4.1 Economic Interpretation

The finding that total volatility dominates prediction has clear economic logic. Volatility measures uncertainty about a company's future. High volatility might indicate uncertain earnings (cyclical businesses, startups), unstable management (frequent turnover), contested competitive position (disruption threats), or regulatory uncertainty. When news arrives such as an earnings announcement, a management change, or a competitor's product launch, uncertain companies experience larger price reactions because the news resolves (or amplifies) existing uncertainty.

The selection of total volatility rather than its decomposition (systematic versus idiosyncratic) is also informative. In principle, systematic volatility (sensitivity to market moves) and idiosyncratic volatility (firm-specific risk) have different economic sources. The data, however, shows they are nearly perfectly correlated (r = 0.99) for most stocks. This means idiosyncratic risk dominates total volatility. The finding then suggests jumps primarily arise from firm-specific shocks (earnings surprises, company-specific news) rather than market-wide moves. Market crashes affect all stocks simultaneously, creating correlated moves, but individual jumps more often reflect idiosyncratic events.

VIX and recession indicators capture macroeconomic conditions that amplify vulnerability. During high-VIX periods, even fundamentally stable firms experience discontinuous price movements. This is not just correlation but a mechanism: market stress deteriorates liquidity (wider bid-ask spreads, less market-making), triggers information cascades (investors sell broadly rather than discriminating), and amplifies reactions to news (heightened uncertainty makes investors more sensitive to information). The recession indicator adds information about economic fundamentals (credit conditions, earnings risk) distinct from market volatility sentiment.

The penny stock indicator's significance reflects structural differences in how low-priced stocks trade. Penny stocks have lower institutional ownership (many funds have minimum price requirements), wider bid-ask spreads relative to price, more retail speculation (unsophisticated traders more prone to overreaction), and higher delisting risk. These factors make penny stocks inherently more fragile beyond what volatility alone captures.

The Finance industry's protective effect (negative coefficient) is interesting. Despite high leverage and interconnectedness, regulated financial firms show lower jump probability after controlling for volatility. This likely reflects regulatory constraints (capital requirements, stress testing, liquidity requirements) that effectively cap tail risk, alongside business model diversification (multiple revenue streams reduce dependence on any single factor).

4.2 Why Simple Models Win

The dominance of linear models over sophisticated alternatives reveals fundamental problem structure. XGBoost's flexibility to capture complex interactions does not help because the underlying relationship is approximately linear. When volatility doubles, jump probability increases in a log-linear fashion. There are no hidden threshold effects ("jumps only occur above volatility X") or higher-order interactions ("volatility and momentum interact in non-obvious ways") that justify complex models.

This has a practical implication: in noisy domains like finance, simple models often outperform complex ones. Complex models have the flexibility to fit training data idiosyncrasies that do not generalize. XGBoost's training AUC of 0.724 substantially exceeds its test AUC of 0.688, a telltale sign of overfitting. Simple linear models maintain tighter train-test correspondence, indicating they capture robust patterns.

Feature engineering matters more than algorithmic sophistication. The choice to include

CAPM-based volatility decomposition, macro stress indicators, and structural firm characteristics (penny stock status) provides the raw material for prediction. Given good features, simple algorithms suffice. Without good features, even sophisticated algorithms struggle.

4.3 LASSO as Optimal Production Model

For practical deployment, LASSO emerges as the optimal choice. It achieves 0.6945 AUC, within 0.0005 of the best model (Ridge). This microscopic performance sacrifice buys substantial practical benefits.

First, simplicity. Five features are trivially easy to compute in real-time, reducing data dependencies and operational complexity. Second, interpretability. The model can be explained to non-technical stakeholders: "We predict jumps using the stock's historical volatility, current market fear level, whether it is a penny stock, whether we are in a recession, and its industry." Third, maintenance. Fewer features mean fewer things that can break, drift, or require monitoring. Fourth, robustness. Sparse models are less prone to degradation when feature distributions shift during regime changes.

The slight AUC edge from Ridge (using all 19 features) or the different error trade-offs from XGBoost are not worth the operational burden for most applications. In production environments where model explainability, regulatory approval, and long-term stability matter, LASSO's simplicity is a feature, not a limitation.

4.4 Practical Applications and Limitations

Portfolio risk managers can use jump predictions for dynamic hedging strategies. Rather than hedging uniformly or based on simple rules (hedge everything above 30% volatility), predictions enable targeted allocation. Hedge the top 30% of predicted jump probabilities if budget allows protecting 30% of the portfolio. This concentrates resources on highest-risk positions.

However, important limitations constrain interpretation. First, these are correlations, not causation. The model identifies that high volatility predicts jumps but cannot definitively prove volatility causes jumps. Both might be driven by underlying fundamental uncertainty. Second, predictions are static monthly estimates. In reality, jump risk evolves intra-month based on news flow. A real-time updating framework would provide more actionable signals. Third, the model predicts binary outcomes (jump or not) rather than magnitude. A 10% move and a 25% move are both classified as jumps, but they have very different portfolio impacts.

Fourth, survivorship bias may understate true risk. The dataset includes only firms that survived their observation period. Delisted firms, often victims of catastrophic price declines, are underrepresented. This may make jumps look more predictable than they truly are. Fifth, the sample period (1996-2023), while including diverse regimes, may not represent future market structures. Algorithmic trading, passive investing, and social media may alter jump dynamics in ways historical data does not capture.

4.5 Future Directions

Improvements should focus on better data rather than more complex models. Alternative data sources such as news sentiment (NLP on earnings calls, media coverage), social media signals (Twitter volume and sentiment, StockTwits activity), and options market signals (implied volatility skew, put-call ratios) might capture information not present in historical prices and macro indicators. High-frequency microstructure data (intraday volatility patterns, order flow imbalance) could provide earlier warning signals. Earnings announcement calendars and analyst forecast dispersion might identify periods of elevated information uncertainty.

Advanced modeling architectures could include LSTMs for temporal dependencies (jumps may cluster for individual stocks), attention mechanisms to weight recent versus distant history, or regime-switching models with different parameters for bull and bear markets. However, given that simple linear models already achieve strong performance, these sophistications should be pursued cautiously with clear evidence they add value.

Causal analysis would strengthen interpretation. Natural experiments such as index rebalancing or regulatory changes could identify causal effects. Instrumental variables might address endogeneity between volatility and jumps. Establishing not just prediction but causation would support more confident policy recommendations.

5 Conclusion

This study demonstrates that stock price jumps, while substantially unpredictable, exhibit meaningful systematic patterns. Across five machine learning approaches trained on 28 years of data covering 1,023 stocks, consistent evidence shows that volatility and macroeconomic stress predict jump probability with AUC around 0.70. This represents meaningful discrimination, allowing models to correctly rank stocks 70% of the time, though it falls far short of perfect prediction.

LASSO identifies five core predictors (total volatility, VIX, penny stock indicator, recession indicator, Finance industry) that capture essentially all available signal. Fourteen other features contribute negligible value, demonstrating that jump prediction has a simple core structure. Simple linear models match or exceed sophisticated alternatives, confirming that the problem involves fundamentally linear relationships.

For practitioners, these findings support data-driven risk management. The modest AUC values (0.70) should not be dismissed. Small systematic edges compound into substantial value when applied consistently across many decisions. The key is understanding when and how to use predictions, acknowledging limitations, and integrating them into comprehensive risk frameworks rather than relying on them exclusively.

The dominance of simple, interpretable models provides an important lesson. In noisy domains like finance, transparency and robustness often matter more than marginal performance gains. LASSO's 5-feature model achieves 99.9% of the best performance while offering vastly superior interpretability, operational simplicity, and likely better resilience to regime changes.

Future work should incorporate richer data sources and develop adaptive models for regime shifts. However, a fundamental challenge will remain: large price movements contain substantial unpredictable components arising from genuine information surprises. No prediction model can fully anticipate the unexpected. The 0.70 AUC ceiling likely reflects this economic reality rather than a methodological failure. Continued innovation in features and methods will incrementally improve jump risk identification, but perfect prediction will remain elusive.

Note: Interactive analysis and detailed methodology available in accompanying Jupyter notebook.

For questions or collaboration opportunities, please contact via portfolio website or LinkedIn.