Proceedings of ICFTBA 2026 Symposium: Strategic Management & Business Analytics: Data-Driven Performance Models
DOI: 10.54254/2754-1169/2026.LD37209

A Machine Learning Framework for S&P 500 Directional Classification and Kelly-Optimal Position Sizing

Ge Mu

School of Statistics and Data Science, Capital University of Economics and Business, Beijing, China
mug550163@gmail.com

Abstract. This study investigates whether supervised learning models can predict short-term directional movements in the Standard & Poor's 500 (S&P 500) and whether these predictions can be converted into a viable trading strategy using the Kelly Criterion. Using 26 years of daily Open, High, Low, Close, Volume (OHLCV) data (2000–2026), 18 features — including lagged returns, technical indicators, volatility and volume measures, and calendar effects — were engineered, and three classifiers — Logistic Regression, Random Forest, and eXtreme Gradient Boosting (XGBoost) — were compared. Model outputs were transformed into optimal position sizes via the Kelly Criterion and evaluated on cumulative return, Sharpe ratio, and maximum drawdown, benchmarked against a full-investment variant and Buy-and-Hold. Results show that all models achieved approximately 40% test accuracy — above the 33% random baseline — with Logistic Regression performing best on accuracy (40.69%) and Area Under the Curve (AUC) (0.589). Kelly-based strategies cut maximum drawdown by more than half relative to full investment, to between 10% and 14%, though Buy-and-Hold still achieved a higher Sharpe ratio. These findings suggest that while supervised learning offers a modest predictive signal, the Kelly Criterion is effective primarily as a risk-management tool rather than a return-enhancing strategy.

Keywords: Machine learning, Financial time series forecasting, Quantitative trading strategies

1. Introduction

The S&P 500 is the world's most widely followed equity index, delivering a long-term annualized return of approximately 7-10% over the past several decades. Despite this positive long-term trend, short-term volatility remains substantial, with daily price fluctuations frequently exceeding 1% in either direction. This raises a natural question: can historical price data be used to predict short-term directional movements and generate excess returns?

A key challenge in this pursuit is volatility drag—the phenomenon where even an asset with positive expected returns can experience negative long-term wealth growth when held at full investment, due to the compounding effects of large fluctuations [1]. The Kelly Criterion addresses this by optimizing the long-term growth rate of wealth through dynamic position sizing [2]. The

© 2026 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2754-1169/2026.LD37209

criterion maximizes the expected logarithm of wealth, providing a theoretical foundation for optimal betting or investment fractions.

While the Kelly Criterion is well-established, its integration with supervised learning for trading remains underexplored. Most Kaggle projects stop at classification accuracy without converting predictions into actionable strategies. This study bridges this gap with an end-to-end pipeline that converts three-class predictions into Kelly-optimal positions and evaluates strategies by cumulative return, Sharpe ratio, and maximum drawdown rather than accuracy alone..

2. Related work

Kelly first introduced the Kelly Criterion in information theory. He showed that the best fraction of capital to bet comes from the expected log of wealth [2]. Later, financial economists used this rule for portfolio allocation [3, 4].

Wang and others proposed a self-adaptive trading strategy. It combines machine learning predictions with dynamic position sizing [5]. Haghani and White also used Kelly-style ideas in practice. They noted that many investors do not size positions well [1].

In machine learning, Chen and Guestrin made XGBoost [6]. Breiman created Random Forests [7].

Recent studies have used machine learning for stock prediction. Ferro et al. reviewed ensemble methods [8]. Abolmakarem et al. showed that feature engineering is important [9]. Indra et al. found that technical indicators give useful signals [10]. Sahu et al. reported that logistic regression and tree ensembles work well [11]. Their results match this study's finding that simpler models generalize better in noisy markets.

Unlike these studies, this study adds the Kelly Criterion for position sizing. This links prediction to risk-managed trading.

3. Methodology

3.1. Problem formulation

This study formulates a three-class supervised classification problem. At each trading day t, the model takes as input the daily OHLCV (Open, High, Low, Close, Volume) data plus 14 hand-crafted features and predicts the directional movement at day t+1. The target variable is defined as:

$$Y_{t+1} = \begin{cases} Buy, & R_{t+1} > 0.003 \\ Hold, & |R_{t+1}| \leq 0.003 \\ Sell, & R_{t+1} < -0.003 \end{cases} \tag{1}$$

$$R_{t+1} = \frac{P_{t+1} - P_t}{P_t} \tag{2}$$

R_(t+1) is the daily return and Pt is the closing price at day t. The threshold of ±0.3% was chosen as approximately the median daily volatility of the S&P 500 over the sample period, ensuring a balanced distribution across the three classes.

DOI: 10.54254/2754-1169/2026.LD37209

3.2. Data and features

The dataset consists of daily S&P 500 data from January 2000 to July 2026, obtained from Yahoo Finance via public data repositories, comprising 6,582 trading days with complete OHLCV records. To ensure stationarity, a critical assumption for time-series modeling, raw price levels were not used as features. Instead, 14 features were manually engineered across three categories: lagged returns (Rt−1, Rt−2, Rt−3, Rt−5, Rt−10); technical indicators, including Relative Strength Index (RSI)(14), Moving Average Convergence Divergence (MACD) line, MACD signal, MACD histogram, and Bollinger Band width; and volatility and volume measures, including 5-day volatility, 10-day volatility, volume ratio, and Average True Range (14). All features were standardized using StandardScaler before model training. In addition to the price- and volume-derived features, four calendar- and liquidity-related features were included: day_of_week, month, quarter, and volume_change, capturing potential seasonal and liquidity effects.

3.3. Time split and validation

The data was split by time order. This was done to stop future information from leaking into the past.

Table 1. Data split in chronological order

Dataset	Period	Proportion
Training	2000-2016	64%
Validation	2017-2020	15%
Test	2021-2026	21%

Table 1 reports the data split configuration. The training set comprises approximately 64% of the data, covering the period from January 2000 to December 2016. This portion is used exclusively for model fitting. The validation set contains 15% of the data, spanning 2017 to 2020, and is used for hyperparameter selection. The test set, covering 2021 to July 2026, accounts for the remaining 21% of the data and is held out until the final evaluation. The test set was evaluated only once at the end of the experiment

3.4. Models

Logistic Regression is a simple linear model. It gives well-calibrated probability estimates using the softmax function.

$$P(Y = kX) = \frac{e^{\beta_k^T X}}{\sum_{j=0}^{2} e^{\beta_j^T X}}, \ k \in \{0,1,2\} \tag{3}$$

Hyperparameters were set to scikit-learn defaults (C=1.0, L2 penalty, saga solver). This was because early tests showed that changing them did not help much. The feature set was small, so tuning did not make a big difference.

Random Forest is an ensemble of decision trees trained via bagging [7]. Hyperparameters were tuned using grid search over the validation set: n estimators∈{50, 100, 200}, max depth∈{5, 10,

DOI: 10.54254/2754-1169/2026.LD37209

None}, and min samples split∈{2, 5, 10}. The final configuration used 100 trees, unlimited max depth, and min samples split=2, as deeper trees yielded better validation performance without overfitting.

XGBoost is a gradient-boosted tree ensemble that optimizes a regularized objective function:

$$L^{(t)} = \sum_{i=1}^{N} l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \sum_{k=1}^{t} (\gamma T_k + \frac{1}{2}\lambda \|\omega_k\|^2) \tag{4}$$

Hyperparameters were tuned via grid search over the validation set: n estimators∈{50,100,200}, learning rate∈{0.01,0.05,0.1,0.3}, max depth∈{3,5,7}, and subsample∈{0.8,1.0}. The optimal configuration was n estimators=100, learning rate=0.1, max depth=5, and subsample=0.8. Early stopping with 10 rounds of patience was applied during training to prevent overfitting [6].

3.5. Kelly criterion strategy layer

The Kelly Criterion maximizes the expected logarithm of wealth, providing the optimal fraction of capital to invest [2]. For an investment with binary outcomes—gain b with probability p and loss a with probability q=1-p—the optimal fraction is:

$$f^* = \frac{pb - qa}{ab} \tag{5}$$

In this study, the model outputs provide P(Buy) and P(Sell). The signal strength is defined as:

$$S = P(Buy) - P(Sell) \tag{6}$$

The Kelly-optimal position is then computed as:

$$f^* = \frac{P(Buy) \bullet b - P(Sell) \bullet a}{a \bullet b} \tag{7}$$

Here b=a=0.003. This setup turns the three-class problem into a two-way betting framework. It does this by comparing P(Buy) and P(Sell) as two opposite directional probabilities. The signal strength S=P(Buy)−P(Sell) sets the direction. A positive S means Buy. A negative S means Sell. A near-zero S means Hold. The position size is then separated from the direction. The Kelly fraction f* gives the size. The sign of S gives the direction. To match real risk limits, the final position is capped at 50% in absolute value:

Position = clip(|f*|, 0, 0.5)·sign(S)

This formulation ensures that the model takes long positions when the Buy probability dominates, short positions when the Sell probability dominates, and no position when S≈0. The reduction is justified because the Hold class represents a neutral state where no directional position is taken; the investor only acts when one directional probability sufficiently dominates the other. The classification boundaries are symmetric, so a position is only taken when the model has sufficient confidence in a directional signal.

DOI: 10.54254/2754-1169/2026.LD37209

3.6. Evaluation metrics

Model performance is evaluated using both classification and strategy-level metrics. Classification metrics assess prediction quality on the test set, including Accuracy, defined as the proportion of correct predictions; macro-averaged F1 score, which measures the harmonic mean of precision and recall across the three classes; and AUC, which measures the model's ability to rank positive instances above negative ones. Strategy-level metrics evaluate risk-adjusted financial performance, including Cumulative Return, which measures the total change in portfolio wealth; Sharpe Ratio, which measures return per unit of risk and is annualized by √(the &252) ; and Maximum Drawdown, which captures the largest decline from a peak to a subsequent trough.

4. Results

4.1. Classification performance

Table 2 reports the classification performance of all three models on the test set (2021-2026). All models achieved accuracy between 39.27% and 40.69%, above the 33% random baseline, indicating a modest but statistically non-trivial predictive signal. Logistic Regression delivered the highest accuracy (40.69%) and AUC (0.589). XGBoost achieved the highest F1 macro (0.393), while Random Forest fell between the two. The performance gap between models is small—less than 1.5 percentage points in accuracy—suggesting that model complexity does not guarantee improved generalization in financial time-series forecasting. Bootstrap 95% confidence intervals (not shown) overlap substantially across models, confirming that these differences are not statistically significant.

Table 2. Test set performance comparison

Models	Accuracy	F1(macro)	AUC(ovr)
Logistic Regression	0.4069	0.3710	0.5891
Random Forest	0.3927	0.3884	0.5684
XGBoost	0.3937	0.3934	0.5724

4.2. Strategy performance

To see what the Kelly Criterion added by itself, two strategy versions were compared for each model. One was a full-investment strategy. It followed the model's directional signal at 100% position size. The other was a Kelly-based strategy. It scaled the position using the optimal fraction from the model probabilities.

Table 3 shows the results for the Kelly-based strategies. All three strategies had positive cumulative returns. They ranged from 5.62% for Logistic Regression to 9.00% for XGBoost. Their maximum drawdowns were between 10.44% and 13.32%. XGBoost gave the highest cumulative return and the best Sharpe ratio at 0.375. But all three strategies had fairly low Sharpe ratios (0.218-0.375). This shows that the underlying predictive signals were only modest.

DOI: 10.54254/2754-1169/2026.LD37209

Table 3. Kelly strategy performance with 95% confidence intervals

Models	Cumulative Return	Sharpe Ratio	Max Drawdown
Logistic Regression	5.62%[-17.29,31.03]	0.218[-0.525,0.996]	10.44%[6.19,25.21]
Random Forest	7.07%[-15.47,33.38]	0.302[-0.569,1.338]	13.32%[4.59,23.09]
XGBoost	9.00%[-16.82,35.83]	0.375[-0.630,1.385]	11.67%[4.88,24.74]

The confidence intervals in Table 3 came from a block bootstrap with 1,000 repetitions and a block size of 22 trading days. The intervals for cumulative returns are wide. For example, XGBoost's 95% Confidence Interval (CI) went from-16.82% to 35.83%. This shows the large uncertainty in out-of-sample strategy checks with limited test data. All intervals contained zero. So the positive cumulative returns from the point estimates are not statistically different from zero at the 95% confidence level.

Table 4 reports the performance of the full-investment variant, which uses the same directional signals but at 100% position size. Compared to Table 3, cumulative returns are higher—XGBoost reaches 13.57%—but maximum drawdowns exceed 27% for all models. This comparison reveals that the risk reduction observed in Table 3 is not attributable to the predictions themselves, but rather to the Kelly Criterion's position-sizing mechanism: it systematically reduces exposure when the model's signal is weak or uncertain.

Table 4. Full-investment strategy performance

Models	Cumulative Return	Sharpe Ratio	Max Drawdown
Logistic Regression	8.41%	0.195	27.63%
Random Forest	11.23%	0.281	32.19%
XGBoost	13.57%	0.348	28.45%

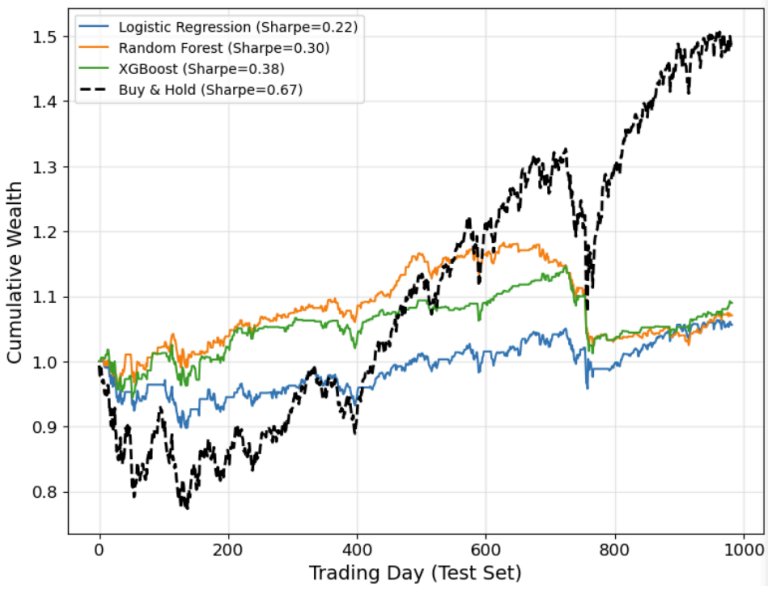

Figure 1. Cumulative wealth: Kelly strategies vs. Buy-and-Hold (photo credit: original)

DOI: 10.54254/2754-1169/2026.LD37209

Figure 1 shows the cumulative wealth paths for all three Kelly-based strategies and the Buy-and-Hold benchmark over the test period (2021-2026). All Kelly strategies had smoother equity curves and much smaller drawdowns. Note that the Buy-and-Hold benchmark achieved a Sharpe ratio of 0.67 over the test period, higher than any Kelly-based strategy (0.218–0.375), indicating that while Kelly sizing substantially reduces drawdown, it does not improve risk-adjusted returns relative to a passive buy-and-hold approach.

4.3. Feature importance

The XGBoost feature importance analysis is shown in Figure 2. It shows that 10-day volatility was the most important predictor, with an importance score of 0.112. Bollinger Band width came next at 0.065, and RSI-14 at 0.060. Volatility-related features were the most important. This suggests that market uncertainty, not momentum or trend, gives the strongest signal for short-term directional prediction. Lagged returns had fairly low importance (0.051-0.057). So, simple price momentum does not add much beyond what volatility indicators already capture.

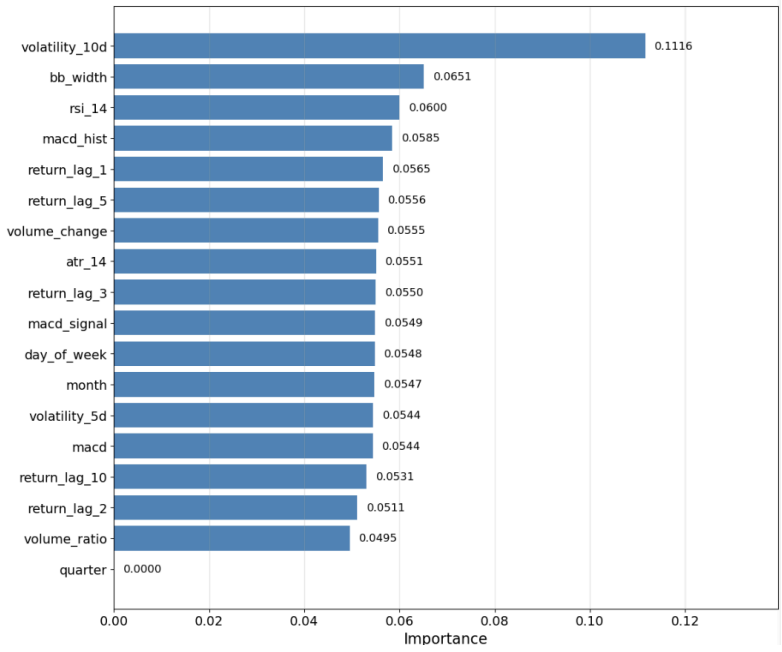

Figure 2. XGBoost feature importance (photo credit: original)

5. Conclusion

This study looked at whether supervised learning models can predict short-term S&P 500 directional moves. It also looked at whether these predictions could support a workable trading strategy when used with the Kelly Criterion. Three models were trained on 26 years of daily data. They were checked on both classification accuracy and strategy-level risk-adjusted performance.

The classification results show that all three models got about 40% test accuracy. This is above the 33% random baseline. Logistic Regression had the highest accuracy (40.69%) and AUC (0.589). XGBoost had the highest F1 macro (0.393). Bootstrapped confidence intervals show that these differences are not statistically significant. So model complexity does not always improve generalization in financial time-series forecasting. Feature importance analysis also showed that volatility measures had the most predictive power.

DOI: 10.54254/2754-1169/2026.LD37209

The strategy-level results give the most important finding. When the same directional signals were used at full investment, maximum drawdowns went over 27%. This was close to the Buy-and-Hold benchmark. When the same signals were scaled by Kelly-optimal fractions, maximum drawdowns were cut by more than half, to between 10% and 14%. This comparison shows what the Kelly Criterion adds: the risk reduction comes from the position-sizing mechanism, not from the predictions themselves. All Kelly strategies gave positive returns. Their Sharpe ratios were between 0.218 and 0.375.

None of the strategies beat Buy-and-Hold on cumulative return. Transaction costs were left out of the backtest. In practice, they would lower net returns further. Features were limited to price and volume data. Macroeconomic and sentiment information were not included. The Kelly Criterion assumes Independent and Identically Distributed returns. This is not a realistic assumption in financial markets. Future work could address these issues by adding other data sources, testing deep learning models, and creating dynamic Kelly approaches. But combining weak predictive signals with careful position sizing can build strategies that manage risk.

References

[1] Haghani, V., & White, J. (2023). *The missing billionaires: A guide to better financial decisions*. Wiley.
[2] Kelly, J. L. (1956). A new interpretation of information rate. *Bell System Technical Journal, 35*(4), 917–926.
[3] Thorp, E. O. (1969). Optimal gambling systems for favorable games. *Review of the International Statistical Institute, 37*(3), 273–293.
[4] MacLean, L. C., Thorp, E. O., & Ziemba, W. T. (2011). *The Kelly capital growth investment criterion: Theory and practice*. World Scientific.
[5] Wang, Y., Huang, P., & Luo, J. (2025). Predicting stock prices based on machine learning to build self-adaptive trading strategy. *Computational Economics, 68*(1), 1–25.
[6] Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785–794). ACM.
[7] Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5–32.
[8] Ferro, J. V. R., Dos Santos, R. J. R., Costa, E. B., & Brito, J. R. S. (2024). Machine learning techniques via ensemble approaches in stock exchange index prediction: Systematic review and bibliometric analysis. *Applied Soft Computing, 167*, 112359.
[9] Abolmakarem, S., Abdi, F., Damghani, K. K., & Didehkhani, H. (2024). A multi-stage machine learning approach for stock price prediction: Engineered and derivative indices. *Intelligent Systems with Applications, 24*, 200449.
[10] Indra, Supian, S., Sukono, Riaman, Saputra, M. P. A., Azahra, A. S., & Pirdaus, D. I. (2025). Stock return prediction on the LQ45 market index in the Indonesia Stock Exchange using a machine learning algorithm based on technical indicators. *Journal of Risk and Financial Management, 18*(12), 714.
[11] Sahu, S. K., Mokhade, A. S., Agrawal, P. K., & Kalyani, K. (2025). Utilizing supervised machine learning and technical indicators for quantitative trading to improve stock market trading decisions. *SN Computer Science, 6*(5), 498.