Data Science in Finance and Economics

https://www.aimspress.com/journal/dsfe

DSFE, 5(4): 466–501.
DOI:10.3934/DSFE.2025019
Received: 04 April 2025
Revised: 13 September 2025
Accepted: 16 October 2025
Published: 29 October 2025

Research article

A forest of opinions: A multi-model ensemble-HMM voting framework for market regime shift detection and trading

Rethyam Gupta[1,*], **Sarthak Kapoor**[2], **Himank Gupta**[3] **and Srinivasan Natesan**[3]

[1] School of Operations Research and Information Engineering, Cornell University, Ithaca, NY 14850, USA
[2] Department of Computer Science and Engineering, Indian Institute of Technology Guwahati, Guwahati, Assam 781039, India
[3] Department of Mathematics, Indian Institute of Technology Guwahati, Guwahati, Assam 781039, India

* **Correspondence:** Email: rg795@cornell.edu.

Abstract: In this paper, we present a framework for detecting market regime shifts using a combination of tree-based ensemble learning models and classical statistical techniques. Specifically, we leverage homogeneous ensemble methods (bagging and boosting) alongside the hidden Markov model (HMM) to identify transitions between different market states (e.g., bull, bear, and neutral). We further propose hybrid voting classifiers that integrate the HMM with specific ensemble learning models to enhance the robustness of regime classification. The model incorporates a comprehensive set of macroeconomic and technical market indicators to provide a holistic view of the underlying market dynamics. Although our primary objective is not to optimize for maximum profitability, we demonstrate that the identified regimes can be utilized effectively to construct a viable trading strategy. Our results, based on exchange-traded funds (ETFs) representing the Russell 3000 and the Standard and Poor's 500 (S&P 500) index, indicate that regime-aware strategies developed through our modeling framework can effectively support informed investment decision-making.

Keywords: hidden Markov model; ensemble learning; voting classifier; market regime shift detection; trading strategies; machine learning in finance

JEL Codes: C52, C53, C58

1. Introduction

Financial markets are characterized by dynamic regimes, each exhibiting distinct behavioral patterns and risk profiles (Hamilton, 1989; Ang and Timmermann, 2012). Timely detection of these regime shifts—such as transitions between bull, bear, and neutral markets—is crucial for effective investment management, risk mitigation, and strategic asset allocation (Guidolin and Timmermann, 2007; Nystrup et al., 2017). Traditionally, financial practitioners and researchers have relied on classical statistical methods, notably hidden Markov models (HMMs) (Baum et al., 1970), to identify these shifts because of their effectiveness in modeling time series data with unobservable states (Hamilton, 1990; Rydén et al., 1998). However, HMMs often suffer from limitations such as sensitivity to parameter initialization, reliance on assumptions of state transition probabilities, and potential overfitting issues in complex financial data environments (Bulla and Bulla, 2006).

Recent advances in machine learning, particularly tree-based ensemble models, have expanded the capabilities for capturing complex, nonlinear interactions within financial datasets. Ensemble methods have been extensively applied in various financial forecasting tasks, demonstrating superior predictive performance compared with traditional linear models (Chen and Guestrin, 2016; Dorogush et al., 2018).

This paper presents a comprehensive framework for detecting market regime shifts by combining classical statistical approaches with machine learning techniques. We integrate ensemble methods—extreme gradient boosting (XGBoost) for boosting (Chen and Guestrin, 2016) and BaggingClassifier for bagging (Breiman, 2001)—with the HMM. The outputs of these models are combined through a voting classifier, which synthesizes their predictions on the basis of predefined rules to produce a final classification. This integration enhances robustness by mitigating individual model limitations and exploiting complementary predictive strengths. Prior studies have demonstrated that such voting mechanisms can improve classification reliability across diverse domains (Kuncheva and Rodríguez, 2012).

Our analysis incorporates an extensive array of macroeconomic and technical market signals, providing a comprehensive representation of market dynamics, an approach aligned with recent literature emphasizing holistic financial modeling (Rapach et al., 2010). Unlike traditional strategies primarily focused on optimizing profitability, our framework emphasizes accurate identification and interpretation of regime transitions. We validate the practical utility of our identified regimes by demonstrating their effectiveness when integrated into a feasible trading strategy. Specifically, we utilize an oracle strategy as the upper bound benchmark, which assumes perfect regime foresight and thus labels the market regimes optimally (Nystrup et al., 2017). Conversely, we adopt a simple buy-and-hold strategy on the index as the lower-bound benchmark. Our regime-aware strategy is evaluated on these bounds using various performance metrics, including the Sharpe ratio (Sharpe, 1966), Sortino ratio (Sortino and Price, 1994), and other relevant industry-standard performance metrics.

By evaluating our models using ETFs that track the Russell 3000 and the S&P 500 index, we demonstrate that a regime-aware trading strategy derived from our proposed voting framework serves as a practical tool for investors seeking alignment between portfolio decisions and prevailing market conditions. The framework offers a more robust risk-adjusted return profile and facilitates more informed investment decisions by integrating the short-term forecasting strengths of ensemble methods with the long-term regime detection capabilities of HMM, as elaborated in Sections 4 and 5. Our work contributes to the existing literature by providing a thorough comparative analysis of ensemble methods and HMM and by introducing a hybrid voting framework explicitly tailored for enhanced regime detection accuracy and practical investment applications.

The work ahead is organized as follows: Section 2 reviews the relevant literature in the domain. Section 3 describes the data preparation process, including how the selected features were engineered and utilized in models. Section 4 outlines the models and methodology employed in the analysis. Section 5 discusses the evaluation metrics and results, and finally, Section 6 concludes the paper with key findings, a discussion on the latest developments in the field, and directions for future work.

2. Literature review

Market regime detection has long been a significant area of focus in financial research due to its crucial role in investment management, asset allocation, and risk mitigation. The earliest methods for identifying market regimes were predominantly based on classical statistical techniques. HMMs, initially introduced by Baum et al. (1970) and popularized in economics by Hamilton (1989), have been extensively utilized due to their ability to model unobservable market states. Despite their widespread adoption, HMMs exhibit several limitations, including sensitivity to the initial parameter values, the restrictive assumption of constant transition probabilities, a tendency to overfit when applied to complex financial datasets, and practical challenges in their implementation (Bulla and Bulla, 2006; Guidolin and Timmermann, 2007; Ang and Bekaert, 2002; Ang and Timmermann, 2012; Abid et al., 2019). In response, more advanced and user-friendly tools have been developed, such as the fHMM R package developed by Oelschläger et al. (2024), which streamlines its implementation on financial time series, hidden state decoding, model checking, selection, and prediction.

As alternatives to HMMs, heuristic and rule-based methods have also been employed extensively by financial media and sell-side research firms to classify market regimes. Traditionally, these heuristics rely on cumulative returns of indices like the S&P 500 to define market conditions. For example, periods with cumulative returns greater than 20% are labeled as bull markets, while those below −20% are classified as bear markets, and ranges between −20% to +5% are classified as neutral markets (Yardeni Research, 2023; Russell Investments, 2020).

To address the constraints of traditional statistical approaches and heuristic methods, recent studies have increasingly employed machine learning models, which offer greater flexibility and capability to capture the nonlinear dynamics inherent in financial markets. Among these, tree-based ensemble methods have emerged prominently due to their robustness and predictive accuracy in forecasting applications. Ensemble methods such as bagging have demonstrated their capability in improving stability and predictive accuracy by aggregating the predictions from multiple weak learners (Breiman, 2001; Patel et al., 2015). Advanced boosting methods like XGBoost further refined ensemble techniques, introducing regularization and improved gradient boosting mechanisms (Chen and Guestrin, 2016).

These models have been particularly effective in financial contexts, addressing common issues like feature selection and overfitting through built-in regularization frameworks, leading to better generalization performance (Fischer and Krauss, 2018; Basak et al., 2019; Zhang et al., 2022; AlQaheri and Panda, 2022). Specifically, Fischer and Krauss (2018) demonstrated significant improvements in predictive accuracy in equity markets by applying ensemble machine learning models. Dixon et al. (2018) employed XGBoost to effectively predict regime shifts in cryptocurrency markets, further validating ensemble methods' capability to handle complex market dynamics. Other works, such as Chen et al. (2019), have also confirmed the effectiveness of boosting algorithms in capturing regime changes and price movements across various asset classes.

While ensemble methods are known to deliver robust predictions individually, integrating their outputs through systematic methodologies has been shown to further improve classification performance. Adam et al. (2024) advanced this line of research by introducing Markov-switching decision trees, which explicitly accommodate regime shifts in the data, while Michels et al. (2023) embedded regression trees within an HMM framework to capture temporal dependence alongside nonlinear relationships. Building on such developments, one promising avenue for combining HMMs with ensemble techniques is the use of voting classifiers, which aggregate predictions from multiple models to mitigate individual weaknesses and amplify collective predictive strength (Polikar, 2006; Kuncheva and Rodríguez, 2012; Zhou, 2012). Despite their potential, such integrative approaches remain underexplored in the domain of market regime detection, thereby presenting a clear research gap that the present study seeks to address.

Historical regime detection methodologies have primarily relied on limited sets of market indicators, predominantly technical indicators or macroeconomic variables, using them independently as features and inputs for the models. Early studies primarily employed basic technical indicators such as exponential moving averages (EMAs) (Murphy, 1999), and moving average convergence divergence (MACD) (Appel, 1979), while others focused primarily on macroeconomic factors like long–term and short–term interest rates, spot prices, and the uncertainty index to detect regime changes (Rapach et al., 2010; Ang and Bekaert, 2002; Chauvet and Hamilton, 2006). More recent literature underscores the importance of adopting a holistic approach that combines macroeconomic and technical features to enhance the robustness of regime detection frameworks (Nystrup et al., 2017; Feng et al., 2018). Nevertheless, extensive integration of these varied feature categories within a single comprehensive framework remains limited despite its potential to improve performance in real-world applications.

Finally, prior research highlights the significance of evaluating detected regimes through practical trading strategies. An oracle strategy (Nystrup et al., 2017; Kritzman et al., 2012), which assumes perfect regime foresight, has often been employed as an upper-bound benchmark in regime detection studies to measure the efficacy of predictive models. Conversely, a simple buy-and-hold strategy typically provides a lower-bound performance benchmark. Relatively few studies have comprehensively evaluated regime detection models using a wide array of industry-standard performance metrics such as the Sharpe ratio, Sortino ratio, maximum drawdown, and annualized and cumulative returns (Feng et al., 2018). Moreover, while the integration of diverse financial indicators has enhanced modeling capabilities, rigorous benchmarking of model performance through trading strategies remains limited, highlighting a promising direction for future research.

3. Data preparation

We utilize an extensive set of financial data and derived indicators from diverse, reputable sources to comprehensively capture market dynamics. Our primary data sources include Yahoo Finance (YFinance), the Federal Reserve Economic Data (FRED) repository, the Chicago Board Options Exchange (CBOE), the US Energy Information Administration (US EIA), and Fineon, an institutional data provider accessed through Cornell University.

We collected data on all relevant features over the period from 2010 to 2025. A walk-forward validation approach (Tashman, 2000) was implemented using a time series split with two folds, resulting in three chronologically ordered segments spanning the years 2010 to 2025. Specifically, the training period from 2010 to 2015 was used to predict market regimes for 2015 to 2020; subsequently, data from 2010 to 2020 were used to forecast the regimes for 2020–2025. Our dataset includes all business days within the specified timeframe. To ensure data continuity and avoid the propagation of missing values, we employed a forward-fill imputation strategy—commonly used in time series analysis to maintain the data's integrity (Tsay, 2010).

YFinance serves as the primary source for market data, such as the Russell 3000 and the S&P 500 index. Macroeconomic indicators like the 3-month Treasury bill rate and 10-year Treasury bond rate are sourced directly from the FRED database. Finally, Fineon, the US EIA, and the CBOE provided detailed data on specialized financial metrics, including the S&P 500 Total Return Index, oil (West Texas Intermediate, WTI) spot price, and the equity put–call ratio, respectively.

Our feature set comprises both technical and macroeconomic indicators, which were deliberately chosen to provide complementary insights into market behavior. Technical indicators are particularly valuable for short-term forecasting, as they effectively capture market sentiment and price momentum (Neely et al., 2014). Consequently, we utilize these technical signals extensively within our ensemble models, as they are known to excel in short-term predictive tasks (Fischer and Krauss, 2018). Among the derived technical indicators, we calculate the MACD, exponential weighted moving averages (EWMAs), detrended price oscillator (DPO), and Bollinger bands, as introduced by Wilder (1978), Appel (1979), Roberts (1959), Murphy (1999), and Bollinger (2001), respectively.

In contrast, macroeconomic indicators have historically demonstrated strong efficacy in modeling longer-term market regimes, particularly through statistical models like the HMM (Hamilton, 1989). Thus, we specifically incorporate macroeconomic data along with some slightly longer-term technical signals in our HMM framework to leverage their documented predictive capabilities in regime detection. The raw and derived macroeconomic indicators utilized include the yield ratio, the oil WTI spot price, the 10-year Treasury bond rate, VIX (volatility index), the 3-month Treasury bill rate, and the New York Stock Exchange (NYSE) short-term trading index (Arms index), each providing additional predictive signals about macroeconomic and market conditions (Ang and Bekaert, 2002; Rapach et al., 2010).

3.1. Technical features, feature tables, and correlation

We construct a set of technical indicators that reflect diverse aspects of price behavior, namely trend direction, volatility, momentum, and investor sentiment. These indicators are derived from Russell 3000 and S&P 500 index data (please note that indicators for both indexes are generated separately). Table 1 provides a description of the indicators used. We further discuss their intuitiveness in detail in Section 4, in the context of feature importance within ensemble methods.

Table 1. Technical indicators used with definitions and modeling intent.

Indicator	Formula/definition	Modeling intent
EMA	$\mathrm{EMA}_t = \alpha P_t + (1-\alpha)\mathrm{EMA}_{t-1}, \alpha = \frac{2}{N+1}$	Capture short-term price trend (Murphy (1999))
EWMA	$\mathrm{EWMA}_t = \lambda r_t^2 + (1-\lambda)\mathrm{EWMA}_{t-1}$	Track recent volatility (Morgan and Reuters (1996))
Daily volatility	$\sigma_t^2 = \dfrac{1}{N-1}\sum(r_{t-i} - \bar{r})^2$	Estimate realized return volatility (Hull (2018))
Bollinger bands	$\mathrm{Upper}_t = \mathrm{SMA}_{20,t} + 2\sigma_t$ $\mathrm{Lower}_t = \mathrm{SMA}_{20,t} - 2\sigma_t$	Overbought/oversold detection (Bollinger (2001))
MACD	$\mathrm{MACD}_t = \mathrm{EMA}_{12,t} - \mathrm{EMA}_{26,t}$ $\mathrm{Signal}_t = \mathrm{EMA}_9(\mathrm{MACD}_t)$	Measure trend momentum (Murphy (1999))
Momentum	$\mathrm{Momentum}_t = P_t - P_{t-N}$	Detect directional acceleration (Murphy (1999))
Average true range (ATR)	$\mathrm{TR}_t = \max(H_t - L_t, \lvert H_t - C_{t-1}\rvert,\ \lvert L_t - C_{t-1}\rvert)$ $\mathrm{ATR}_t = \dfrac{1}{N}\sum \mathrm{TR}_{t-i}$	Capture volatility from price gaps (Wilder (1978))

Continued on next page

Table 1 – continued from previous page

Indicator	Formula/definition	Purpose
Daily range	$\text{Range}_t = H_t - L_t$	Measure intraday price spread (Hull (2018))
DPO	$\text{DPO}_t = P_{t-(\frac{n}{2}+1)} - \text{SMA}_n(P_t)$	Highlights short-term cycles by removing longer-term trends (Murphy (1999))
Put/call ratio	$\text{PCR}_t = \frac{\text{put volume}}{\text{call volume}}$	Capture investor sentiment (CBOE (2023))
Put/call ratio (lagged)	$\text{PCR}_{t+1} = \text{PCR}_t$	Capture investor sentiment (CBOE (2023))

Symbol definitions for table 1: $\alpha = 2/(N+1)$ is the smoothing constant used in the EMA calculation; λ is the smoothing parameter in EWMA; r_t denotes the return at time t; $\bar{r}$ is the average return; σ_t is the estimated daily volatility; H_t, L_t, and C_t represent the high, low, and closing prices at time t, respectively; P_t and P_{t-N} denote the current and N-period lagged prices; and $SMA_{k,t}$ is the k-period simple moving average at time t.

Our features for HMM and ensemble models are presented in Table 2 and Table 3 , clearly indicating whether each feature is directly sourced (raw) or derived through transformations (derived), along with their original sources and the models used for.

Table 2. Macroeconomic – features, sources, types, and models used for them.

Feature	Source	Type	Decay	Ensemble	HMM
Oil WTI spot price	US EIA	Raw	-	-	✓
Yield ratio	FRED	Derived	-	-	✓
VIX	YFinance	Raw	-	-	✓
Arms index	FRED	Raw	-	-	✓
3 month treasury bill rate	FRED	Raw	-	-	✓
10 year treasury bond rate	FRED	Raw	-	-	✓

Table 3. Technical – features, sources, types, and models used for them. The following features are generated separately for the S&P 500 and the Russell 3000.

Feature	Source	Type	Decay	Ensemble	HMM
Daily range	YFinance	Derived	-	-	✓
14 day EMA	YFinance	Derived	14 days	-	✓
1 day lagged equity put call ratio	CBOE	Raw	1 day	-	✓
Equity put call ratio	CBOE	Raw	-	✓	✓
20 day EWMA	YFinance	Derived	20 days	✓	✓
MACD main line	YFinance	Derived	26 days	✓	✓
Daily volatility	YFinance	Derived	-	✓	-
50 day EWMA	YFinance	Derived	50 days	✓	-
MACD signal line	YFinance	Derived	9 days	✓	-
Bollinger upper band	YFinance	Derived	20 days	✓	-
Bollinger lower band	YFinance	Derived	20 days	✓	-

Continued on next page

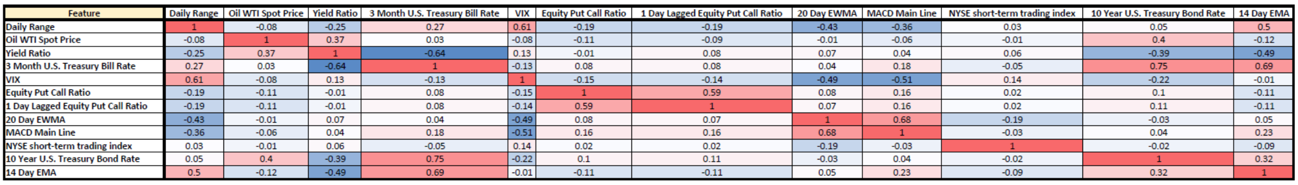
Table 3 (Continued)

Feature	Source	Type	Decay	Ensemble	HMM
14 day momentum	YFinance	Derived	14 days	✓	-
14 day ATR	YFinance	Derived	14 days	✓	-
DPO	YFinance	Derived	20 days	✓	-
S&P total return index	Finaeon	Raw	-	✓	-

Unlike traditional regression models, which can yield unstable coefficient estimates under multicollinearity (Kutner et al., 2005), ensemble-based approaches such as XGBoost and BaggingClassifier exhibit a greater capacity to accommodate highly correlated predictors. XGBoost employs a gradient-boosted decision tree framework with inherent regularization and greedy feature-splitting strategies, selecting the most informative variables at each node while implicitly down-weighting redundant predictors (Chen and Guestrin, 2016). Bagging methods mitigate the influence of correlated features by training multiple base learners on bootstrapped samples and averaging their predictions, thereby reducing variance and the dominance of any single variable (Breiman, 2001).

In contrast, HMM is inherently more sensitive to high inter-feature correlation. While the HMM framework does not impose explicit restrictions on correlated observables, strong collinearity can lead to emission distributions that are either overly peaked or insufficiently distinct across hidden states, thereby impairing state discrimination. This limitation is conceptually analogous to the challenges posed by multicollinearity in regression, where high predictor correlation inflates the standard errors and reduces a model's interpretability (Dormann et al., 2013). Furthermore, unlike ensemble methods, classical HMM formulations lack built-in mechanisms for feature selection or regularization.

Best-practice heuristics in statistical modeling suggest that pairwise correlations above approximately 0.80 can adversely impact model stability, whereas values in the range of 0.40–0.60 are generally more manageable. In constructing our HMM feature set, we aimed to retain variables with low mutual correlations while ensuring the interpretability of the selected features. The highest observed pairwise correlation in our HMM input space is 0.75, occurring between the 3-month and 10-year yield series. Despite this relatively high correlation, including both maturities is important, as they capture different segments of the yield curve—short-term policy-driven dynamics versus long-term expectations on growth and inflation—which provide complementary information for regime identification. To promote transparency, Figure 1 presents the full correlation matrix for all features utilized in the HMM model.



Feature	Daily Range	Oil WTI Spot Price	Yield Ratio	3 Month U.S. Treasury Bill Rate	VIX	Equity Put Call Ratio	1 Day Lagged Equity Put Call Ratio	20 Day EWMA	MACD Main Line	NYSE short-term trading index	10 Year U.S. Treasury Bond Rate	14 Day EMA
Daily Range	1	-0.08	-0.25	0.27	0.61	-0.19	-0.19	-0.43	-0.36	0.03	0.05	0.5
Oil WTI Spot Price	-0.08	1	0.37	0.03	-0.08	-0.11	-0.09	-0.01	-0.06	-0.01	0.4	-0.12
Yield Ratio	-0.25	0.37	1	-0.64	0.13	-0.01	0.08	0.07	0.04	0.06	-0.39	-0.49
3 Month U.S. Treasury Bill Rate	0.27	0.03	-0.64	1	-0.13	0.08	0.08	0.04	0.18	-0.05	0.75	0.69
VIX	0.61	-0.08	0.13	-0.13	1	-0.15	-0.14	-0.49	-0.51	0.14	-0.22	-0.01
Equity Put Call Ratio	-0.19	-0.11	-0.01	0.08	-0.15	1	0.59	0.08	0.16	0.02	0.1	-0.11
1 Day Lagged Equity Put Call Ratio	-0.19	-0.11	-0.01	0.08	-0.14	0.59	1	0.07	0.16	0.02	0.11	-0.11
20 Day EWMA	-0.43	-0.01	0.07	0.04	-0.49	0.08	0.07	1	0.68	-0.19	-0.03	0.05
MACD Main Line	-0.36	-0.06	0.04	0.18	-0.51	0.16	0.16	0.68	1	-0.03	0.04	0.23
NYSE short-term trading index	0.03	-0.01	0.06	-0.05	0.14	0.02	0.02	-0.19	-0.03	1	-0.02	-0.09
10 Year U.S. Treasury Bond Rate	0.05	0.4	-0.39	0.75	-0.22	0.1	0.11	-0.03	0.04	-0.02	1	0.32
14 Day EMA	0.5	-0.12	-0.49	0.69	-0.01	-0.11	-0.11	0.05	0.23	-0.09	0.32	1

Figure 1. The majority of the pairwise correlations are near zero or negative, while the remaining values lie much lower than the commonly accepted threshold of approximately 0.5, indicating that the features are sufficiently distinct from one another.

4. Methodology

In this section, we present a comprehensive framework for detecting market regime shifts and translating those insights into an actionable trading strategy. We begin by outlining the oracle labels, which assume perfect future hindsight for regime detection, then we dive deep into ensemble methods and the HMM, each offering distinct strengths in capturing market dynamics. We further propose a voting mechanism that integrates the outputs of these models, mitigating the limitations of any single approach and improving the overall regime classification accuracy.

The framework is built upon a diverse set of macroeconomic and technical indicators, as discussed in Section 3, selected to provide a well-rounded representation of the underlying market conditions. Once regimes are classified into bullish, bearish, or neutral states, we implement a simple regime-aware trading strategy that adjusts the positions on the basis of the predicted state. This strategy is then benchmarked against individual models, a buy-and-hold baseline, and a strategy with oracle labels, using industry standard metrics discussed in detail in Section 5.

4.1. Oracle labels

To build our models, we utilize a target benchmark referred to as oracle labeling. This labeling methodology assumes perfect foresight of future market regimes and is constructed using oracle labels generated in hindsight. Specifically, market states are classified as bullish or bearish if the percentage change in the index—measured from the most recent peak to the trough (or vice versa)—exceeds a threshold of 15%, a convention consistent with prior studies in financial literature (Nystrup et al., 2017; Rapach et al., 2010). To mitigate the impact of short-term volatility and avoid abrupt transitions between regimes, we introduce a third category, neutral, defined as a buffer zone of seven trading days on either side of a regime shift. To avoid lookahead bias, oracle labels are shifted forward so that predictions for the regime on Day $t + 1$ are based solely on information available up to Day t. As such, the oracle labels serve as an idealized benchmark, representing the maximum achievable performance for any model-based approach under perfect regime identification.

4.2. Ensemble methods

Ensemble methods aggregate multiple model predictions to improve predictive accuracy, robustness, and stability by mitigating the weaknesses inherent to individual models (Dietterich, 2000). By combining diverse predictive models, ensembles achieve lower variance, reduced overfitting, and enhanced generalization capabilities. Broadly, ensemble techniques can be categorized into two key frameworks: Boosting, which combines weak learners sequentially to correct previous errors (Freund and Schapire, 1997), and bagging, which aggregates predictions from independently trained models to reduce variance (Breiman, 2001).

4.2.1. Boosting and extreme gradient boosting (XGBoost)

Boosting constructs powerful predictive models by sequentially combining weaker learners, each attempting to correct the errors of previous iterations. Initially introduced by Schapire (1990) and later enhanced by Freund and Schapire (1997), boosting assigns weights to training instances, emphasizing observations misclassified by earlier models, thereby progressively minimizing prediction errors.

A very popular development of the boosting mechanism, extreme gradient boosting, more commonly called XGBoost, developed by Chen and Guestrin (2016), refines boosting by introducing regularization terms to control the model's complexity, significantly improving generalization and preventing overfitting. Unlike classical boosting algorithms like AdaBoost (Freund and Schapire, 1997) or gradient boosting machines (Friedman, 2001), XGBoost incorporates both L1 (lasso) (Tibshirani, 1996) and L2 (ridge) (Hoerl and Kennard, 1970) regularization in its objective to improve generalization and prevent overfitting, enabling it to control overfitting more effectively. The regularized objective function is

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t), \tag{1}$$

$$\Omega(f_t) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2, \tag{2}$$

where l is the loss function, f_t is the newly added function (decision tree) at iteration t, T is the number of leaves, w_j is the score on the j^{th} leaf, and γ, λ are regularization parameters. Using a second-order Taylor expansion, the model minimizes

$$\mathcal{L}^{(t)} \approx \sum_{i=1}^{n} \left[g_i f_t(x_i) + \frac{1}{2} h_i f_t(x_i)^2 \right] + \Omega(f_t), \tag{3}$$

where $g_i = \partial_{\hat{y}^{(t-1)}} l(y_i, \hat{y}^{(t-1)})$ and $h_i = \partial^2_{\hat{y}^{(t-1)}} l(y_i, \hat{y}^{(t-1)})$ are the first and second-order gradients, respectively. This second-order optimization, combined with regularization, yields a more stable and accurate model compared with classical boosting methods. XGBoost has demonstrated superior performance in forecasting market trends, trading signals, and volatility dynamics (Basak et al., 2019; Shi et al., 2022).

4.2.2. Bagging (bootstrap aggregation) and BaggingClassifier

Bagging, proposed by Breiman (2001), enhances predictive accuracy by averaging predictions from multiple models trained independently on bootstrap samples, thereby reducing variance without substantially increasing bias. The aggregated prediction from the BaggingClassifier is expressed as

$$\hat{y} = \text{argmax}_y \sum_{i=1}^{B} I(h_i(x) = y), \tag{4}$$

where $h_i(x)$ is the prediction of the i^{th} base classifier trained on the i^{th} bootstrap sample. Bagging mathematically reduces variance through averaging, maintaining low bias, and its generalization error benefits from the independence of the base models, with variance reduction approximated by

$$\text{Var}(\hat{f}(x)) = \frac{1}{B^2} \sum_{i=1}^{B} \text{Var}(h_i(x)) + \text{covariance terms}. \tag{5}$$

The BaggingClassifier differs from other bagging methods like random forests by allowing base learners beyond decision trees, leading to better handling of complex data structures (Khan et al., 2024). Its demonstrated capability in reducing overfitting and enhancing predictive stability makes it highly suitable for market regime detection under uncertain conditions (Zhou, 2012).

4.2.3. Training of ensemble methods

Our methodology is designed to ensure robust model performance, reduce overfitting, and enhance generalization to unseen financial data. To achieve this, we incorporate a systematic approach combining walk-forward validation, thorough hyperparameter tuning, pipeline-based preprocessing, and final model deployment. Given the temporal dependencies inherent in financial data, we employ walk-forward validation (Pardo, 2006) based on time series cross-validation methodologies, as outlined by Tashman (2000).

Specifically, we utilize a two-fold cross-validation, where each iteration chronologically segments the dataset into training and testing subsets. Crucially, training sets always precede test sets in time, thereby maintaining the temporal structure and emulating realistic scenarios of financial decision-making and model updating. Figure 2 visually illustrates our walk-forward validation strategy. In Fold 1, the training data span from 2010 to 2015, with the subsequent five-year period (2015 to 2020) designated as the test set. Fold 2 expands the training data period to cover 2010 to 2020, followed by testing on the data from 2020 to 2025. This incremental approach reflects real-world financial forecasting practices, where models are periodically updated as new data become available. To strictly prevent information leakage, each training fold generates a distinct, fold-specific label mapping solely based on historical data available up to that point in time.

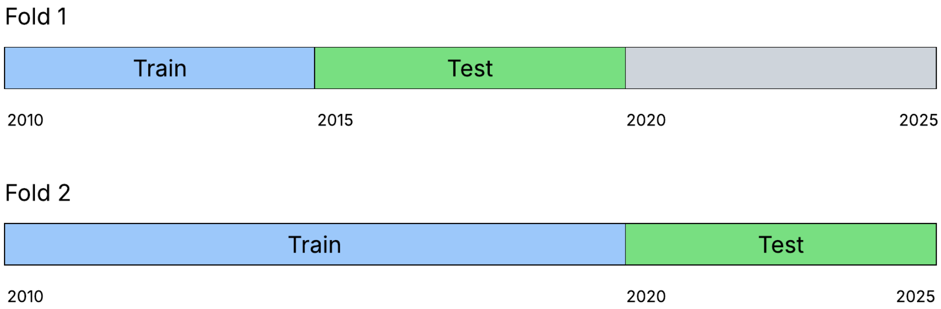


Figure 2. Walk-forward validation mechanism illustrating two folds. In Fold 1, training is done on 2010–2015 and testing on 2015–2020. In Fold 2, the training window extends to 2020, with testing on 2020–2025.

Each test set in the walk-forward validation is strictly out-of-sample, serving as a proxy for real-world performance evaluation, as the model is trained exclusively on prior data and never reuses test samples in future iterations. No separate holdout set is used beyond the walk-forward test sets, as each fold's test segment sufficiently reflects realistic out-of-sample evaluation conditions due to strict chronological partitioning and single-use testing per period.

Given the complex and nonlinear nature of financial markets, hyperparameter optimization is critical to ensure robust predictive performance. We utilize RandomizedSearchCV (Bergstra and Bengio, 2012) to efficiently explore the hyperparameter space. A simplified working mechanism of RandomizedSearchCV is illustrated in Figure 3. This optimization further incorporates TimeSeriesSplit (Pedregosa et al., 2011) with the number of splits being 3, ensuring proper sequential data partitioning that is suitable for time series forecasting.

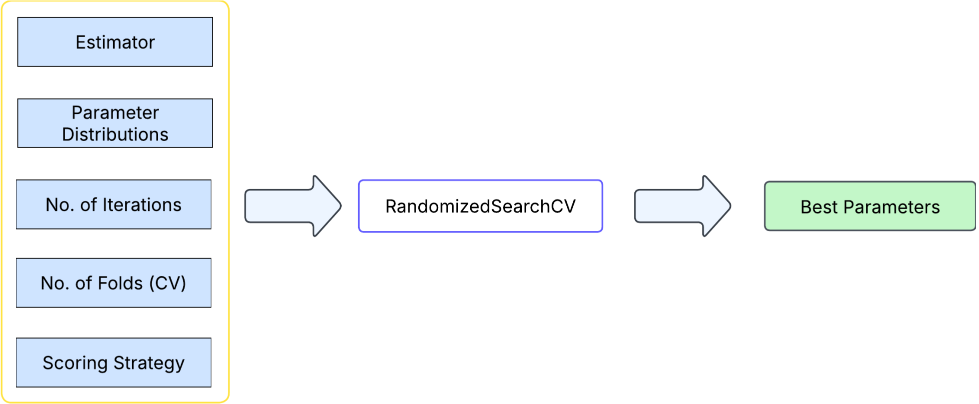


Figure 3. A simplified working mechanism of RandomizedSearchCV, which samples random combinations of hyperparameters and evaluates them using cross-validation to identify the best-performing parameter set.

In our boosting approach, XGBoost is employed, and parameters including maximum depth, learning rate, number of estimators, gamma, regularization alpha, and regularization lambda are chosen using RandomizedSearchCV. Conversely, for the bagging approach, we implement a BaggingClassifier with a DecisionTreeClassifier (Quinlan, 1986) as its base estimator. Here, parameters such as the number of estimators, maximum samples, maximum features, and the base estimator's maximum depth and minimum sample split are optimized. We select balanced accuracy as our evaluation metric to mitigate the adverse effects of class imbalance, ensuring reliable predictions across diverse market regimes (He and Garcia, 2009).

A processing pipeline is constructed for each fold within the walk-forward validation scheme. Initially, all input features are standardized using StandardScaler (Pedregosa et al., 2011) to normalize their scales and prevent biases arising from differences in the features' magnitudes (Hastie et al., 2009). The standardized features are subsequently fed into the ensemble models. For the boosting framework, XGBoost is configured with a multi-class objective function corresponding to the number of regime classes identified in the training data. For the bagging framework, predictions are generated using the BaggingClassifier with the base decision tree as the base classifier. After training, predictions for the test split are mapped back to the original regime labels through an inverse mapping.

The performance of each model is evaluated using various quantitative metrics relevant to financial applications to provide comprehensive insights into the model's effectiveness and risk-adjusted returns, enabling an informed assessment of the performance of the generated signals, which are subsequently used to develop a viable trading strategy, as discussed in detail in Section 5.

Discussion of ensemble method features and correlation: The presence of high correlation among technical indicators is a common characteristic in financial datasets, particularly when multiple features are derived from overlapping price series. Unlike traditional linear models that are sensitive to multicollinearity and may produce unstable coefficient estimates, ensemble-based algorithms such as XGBoost and BaggingClassifier are inherently robust to such issues, as discussed in Section 3 as well. These methods inherently reduce the adverse effects of feature redundancy and allow the models to extract meaningful structure even from highly correlated input spaces (Chen and Guestrin, 2016).

To assess the relative contribution of each feature, we compute importance scores using the respective methodologies of each ensemble model. In XGBoost, the F score reflects how frequently a feature is selected in decision tree splits, whereas in BaggingClassifier, importance is derived from the average contribution across the ensemble of trees. Across both models, we observe that the feature importance scores for all included variables exceed the commonly accepted interpretability threshold in the applied finance literature (Fischer and Krauss, 2018), underscoring their relevance. This empirical result aligns with financial intuition, as each technical indicator encodes a distinct aspect of market behavior and investor sentiment.

Volatility-based features such as the daily range and 14-day ATR capture short-term uncertainty and the magnitude of intraday or multi-day price movements. Momentum indicators like the 14-day EMA, MACD, and 14-day momentum reflect the persistence of price trends, helping to identify directional movement and trend continuation. Mean-reversion features, including Bollinger bands and DPO, provide signals for potential turning points. Sentiment-related variables, such as the equity put/call ratio and its 1-day lagged variant, offer insight into prevailing investor sentiment. Lastly, broader market performance measures like the S&P total return index serve as a macro-level anchor, enabling the model to contextualize firm-level or sector-specific signals. Collectively, these indicators form a comprehensive feature set that spans short- and long-term horizons, directional and volatility-based metrics, and both price-based and sentiment-driven components.

4.3. *The Hidden Markov Model*

HMMs, introduced by Baum et al. (1970), are stochastic models that are adept at handling sequential data characterized by unobserved (hidden) states. Hamilton (1989) significantly advanced their financial applications by modeling economic regime shifts. Mathematically, HMMs assume a Markov property, indicating that the probability of transitioning to the next state depends solely on the current state, formalized as

$$P(q_{t+1} = S_j | q_t = S_i, q_{t-1}, \ldots, q_1) = P(q_{t+1} = S_j | q_t = S_i) = a_{ij}, \tag{6}$$

where a_{ij} denotes transition probabilities between states. Observations are generated from hidden states through emission probabilities $b_j(o_t) = P(o_t | q_t = S_j)$, with parameters typically estimated using expectation maximization (EM) algorithms, notably the Baum–Welch algorithm (Rabiner, 1989). In finance, HMMs have successfully detected regime transitions in equity markets, bond markets, and cryptocurrency markets (Hassan and Nath, 2005; Nguyen and Nguyen, 2015). However, their major limitations—sensitivity to the initial parameters, stationary state-transition probabilities, and potential overfitting—motivate the exploration of ensemble methods.

4.4. *Motivation and regime states*

Financial markets are driven by heterogeneous dynamics that manifest as periods of sustained growth, decline, or stagnation. These dynamics are naturally modeled as latent regimes, since the underlying market condition is not directly observable but leaves statistical footprints in observable variables. We model these dynamics with an HMM with three hidden states: bullish market ($S_t = +1$), neutral narket ($S_t = 0$), and bearish market ($S_t = -1$). This tripartite regime structure is motivated by both domain knowledge and empirical evaluation. In financial economics, market cycles are commonly segmented into bullish, bearish, and neutral periods, reflecting broad investor sentiment and macroeconomic trends.

These hidden states drive observable market outcomes such as price rises, fluctuations, or falls, as previously described in our discussion on oracle labels.

4.5. Mathematical structure of the HMM

Hidden state dynamics: We model the latent market regimes $\{S_t\}_{t=1}^{T}$ using a first-order Markov chain, where each state represents a distinct structural regime. The transition probabilities $a_{ij} = P(S_t = j \mid S_{t-1} = i)$ define the likelihood of switching between regimes and are encoded in a $K \times K$ transition matrix A. This structure captures both persistent and transitory dynamics in financial environments.

Observation model and Gaussian emissions: At each time step t, the model observes a d-dimensional vector $\mathbf{x}_t \in \mathbb{R}^d$ of market features. Conditional on the latent state $S_t = j$, we assume the following

$$\mathbf{x}_t \mid (S_t = j) \sim \mathcal{N}(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) \tag{7}$$

This multivariate Gaussian framework allows for flexible modeling of the joint behavior of continuous features within each regime. The assumption is justified by the empirical observation that many normalized financial market variables, especially those related to volatility and momentum, exhibit approximately elliptical symmetry and regime-dependent variation (Guidolin and Timmermann, 2006). Furthermore, the Gaussian emission model facilitates efficient parameter estimation via expectation maximization and closed-form likelihoods, which are well-suited for high-dimensional financial data.

While the Gaussian assumption is a practical default, it may understate tail risks. Alternative emission distributions, such as the Student's-t (Dias and Embrechts, 2010), have been proposed to capture heavy-tailed phenomena in financial series. We acknowledge these extensions and view them as promising directions for future refinement.

Target variable and regime label mapping: The HMM does not directly predict an observable target in the supervised learning sense. Instead, it infers latent regime states from the joint dynamics of market features. To associate these hidden states with interpretable market regimes ("bull," "bear," "neutral"), we define a directional reference signal—termed the "oracle"—which categorizes market movements according to the sign and magnitude of returns over a fixed horizon. After fitting the HMM, we apply the Viterbi algorithm (Viterbi, 1967) to decode the most likely sequence of latent states over the test period. We then align each decoded state with a regime label by comparing it with the corresponding oracle values: For each latent state, we assign the label (e.g., +1, 0, or –1) that appears most frequently among the time points it occupies. This majority voting procedure establishes a one-to-one mapping from HMM states to economically interpretable regimes, ensuring consistency across validation folds and enabling clear downstream analysis.

While HMMs have been extensively used to model volatility regimes in financial markets, distinguishing between tranquil and turbulent periods on the basis of the changes in variance or covariance structures (Ang and Bekaert, 2002; Guidolin, 2011), they have also been applied to capture shifts in the mean levels of economic indicators and financial signals. Our modeling choice reflects this broader perspective: Financial regimes are characterized not only by differences in volatility but also by persistent structural changes in levels of key variables, such as rising momentum, widening spreads, or changing put-call ratios across bullish, neutral, and bearish phases. Capturing regime-specific means enables the model to identify and interpret such directional shifts, thereby providing a more comprehensive understanding of latent market dynamics.

Complete parameter set, likelihood and inference: The model is parameterized by

$$\Theta = \left(\pi, A = [a_{ij}], B = [b_{jk}] \text{ or equivalently } \{\mu_j, \Sigma_j\}_{j=1}^{3}\right). \tag{8}$$

where π is the initial state distribution.

The likelihood of observing a sequence $\mathbf{x}_{1:T}$ is

$$p(\mathbf{x}_{1:T} \mid \Theta) = \sum_{s_{1:T}} \pi_{s_1} b_{s_1}(\mathbf{x}_1) \prod_{t=2}^{T} a_{s_{t-1},s_t}\, b_{s_t}(\mathbf{x}_t), \tag{9}$$

where the sum runs over all possible hidden state sequences $s_{1:T}$.

The model integrates over every possible hidden regime path, weighting by (i) the probability of following that path through the transition matrix A, and (ii) the likelihood of generating the observed features under each regime's emission distribution.

4.6. Parameter estimation: Baum–Welch algorithm

Directly maximizing the likelihood of an HMM is intractable, since it requires summing over all possible hidden state sequences, which grows exponentially with the sequence length T. To address this, parameters are estimated via the EM algorithm, also known as the Baum–Welch procedure (Rabiner, 1989). The algorithm alternates between computing the probabilities of being in each hidden state (the expectation-step) and re-estimating the model parameters using these probabilities (the maximization-step). This process is guaranteed to increase the likelihood at each iteration.

In the expectation step, we compute two sets of quantities using the forward–backward algorithm

$$\gamma_t(j) = \Pr(S_t = j \mid \mathbf{x}_{1:T}, \Theta^{\text{old}}), \quad \text{(probability of being in state } j \text{ at time } t\text{);} \tag{10}$$

$$\xi_t(i,j) = \Pr(S_t = i, S_{t+1} = j \mid \mathbf{x}_{1:T}, \Theta^{\text{old}}), \quad \text{(probability of transitioning } i \to j \text{ at } t\text{).} \tag{11}$$

These quantities can be interpreted as "soft" assignments of each time step to each regime and of each pair of consecutive time steps to a regime transition.

In the maximization step, the parameters are updated according to the expectations computed above

$$\pi_j^{\text{new}} = \gamma_1(j), \quad \text{(initial probability of starting in state } j\text{);} \tag{12}$$

$$a_{ij}^{\text{new}} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)}, \quad \text{(expected transitions } i \to j \text{ over expected time in } i\text{);} \tag{13}$$

$$\mu_j^{\text{new}} = \frac{\sum_{t=1}^{T} \gamma_t(j)\, \mathbf{x}_t}{\sum_{t=1}^{T} \gamma_t(j)}, \quad \text{(weighted mean of features in state } j\text{);} \tag{14}$$

$$\Sigma_j^{\text{new}} = \frac{\sum_{t=1}^{T} \gamma_t(j)\, (\mathbf{x}_t - \mu_j)(\mathbf{x}_t - \mu_j)^{\top}}{\sum_{t=1}^{T} \gamma_t(j)}. \quad \text{(weighted covariance of features in } j\text{).} \tag{15}$$

The expectation step infers how likely each observation is to belong to each hidden regime and how likely transitions between regimes occur. The maximization step then redefines each regime in statistical terms.

1. The transition matrix $A = [a_{ij}]$ is updated by dividing the expected number of $i \to j$ transitions by the expected total time spent in i,

2. The regime-specific mean μ_j is updated as the weighted average of all observations, where the weights are the probabilities of being in state j, and
3. The covariance Σ_j captures how volatile and correlated the features are within that regime.

Through repeated iterations, the algorithm refines both the regime's definitions and the transition structure until the parameter estimates stabilize. Conceptually, the algorithm alternates between "guessing" the hidden regime assignments, given the current parameters, and "re-estimating" the parameters given those assignments.

4.7. Decoding regimes

After training, the most likely state sequence is decoded using the Viterbi algorithm (Viterbi, 1967)

$$\hat{S}_{1:T} = \arg\max_{s_{1:T}} P(s_{1:T} \mid \mathbf{x}_{1:T}, \hat{\Theta}). \tag{16}$$

This dynamic programming method avoids enumerating all paths and finds the optimal regime sequence efficiently.

Because HMM states are permutation-invariant (e.g., the state "1" may correspond to a bull regime in one run and a bear regime in another), we align the decoded states $\hat{S}_t$ with the oracle labels $O_t \in \{-1, 0, +1\}$ using the Hungarian algorithm (Kuhn, 1955) utilizing confusion counts as shown below

$$C_{j\ell} = \sum_{t=1}^{T} \mathbf{1}\{\hat{S}_t = j, O_t = \ell\}. \tag{17}$$

Find the permutation $\pi^\star$ that maximizes the overlap

$$\pi^\star = \arg\max_{\pi} \sum_{j=1}^{3} C_{j,\pi(j)}. \tag{18}$$

The aligned regimes are then

$$\widetilde{S}_t = \pi^\star(\hat{S}_t) \in \{-1, 0, +1\}. \tag{19}$$

This ensures the regimes produced by the model are labeled consistently as "bull", "neutral", or "bear" in accordance with the oracle.

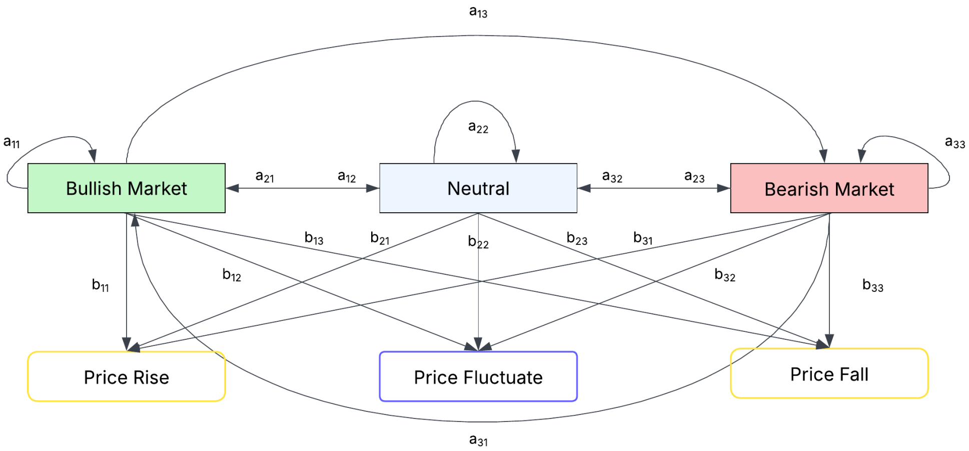


Figure 4. Hidden states (bullish, neutral, bearish) transition with probabilities a_{ij}. Each hidden state generates observable outcomes ("price rise", "price fluctuate", "price fall") with emission probabilities b_{jk}. This three-state HMM diagram illustrates the two layers of stochasticity: at the top level, the regime evolves over time according to A; At the bottom level, each regime emits observable market outcomes with probabilities defined by B.
While the figure does not mathematically capture the full state assignment process of an HMM, it provides an intuitive surface-level idea of how regimes and observations are connected, suitable for readers not seeking a deep technical treatment

.

4.8. *Voting classifiers*

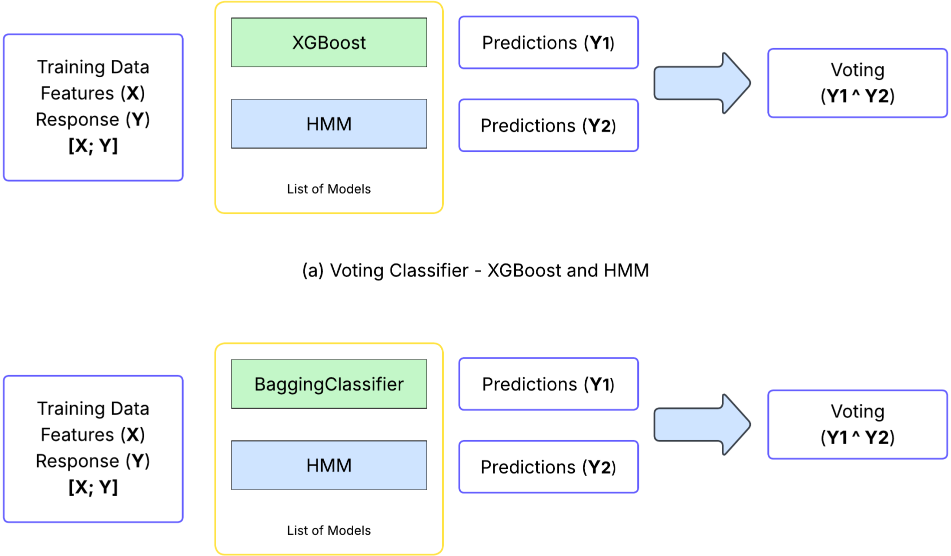
Finally, after implementing the ensemble methods and HMM model, we implement two distinct voting classifiers. The first classifier integrates the predictions from an XGBoost model with the HMM. The second classifier similarly integrates a BaggingClassifier with the HMM. A voting classifier aggregates predictions according to the following

$$\hat{y} = \text{argmax}_y \sum_{j=1}^{J} w_j I(h_j(x) = y), \tag{20}$$

where w_j represents the weights assigned to individual models on the basis of their predictive performance, and $h_j(x)$ denotes the prediction from the j^{th} classifier. Voting classifiers leverage collective strengths while compensating for individual model weaknesses, resulting in more reliable and stable predictions. This approach has demonstrated superior performance in various financial tasks, including regime detection, volatility modeling, and portfolio optimization (Kuncheva and Rodríguez, 2012; Zhou, 2012; Polikar, 2006; Ballings et al., 2015; Nystrup et al., 2017; Fischer and Krauss, 2018; Krauss et al., 2017; De Prado, 2018; AlQaheri and Panda, 2022; Henrique et al., 2019).

The voting classifier architecture, depicted explicitly in Figure 5, and the voting strategy, shown in Figure 6, classifies the market condition as bull, bear, or neutral on the basis of combined predictions (Y_1 from XGBoost or BaggingClassifier, and Y_2 from HMM). We define two classification types for combining predictions. The "aggressive strategy" assigns a bullish or bearish label even if only one classifier strongly indicates it, unless directly contradicted, making it more responsive but risk-prone.

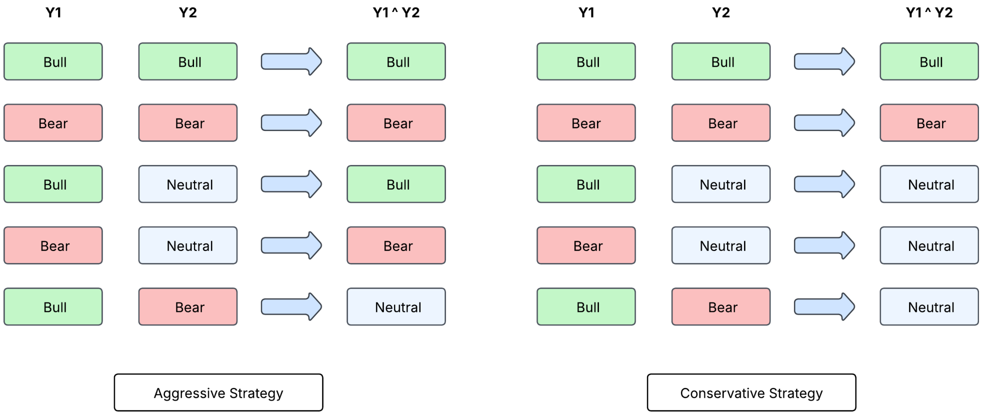
In contrast, the “conservative strategy” defaults to neutral unless there is clear agreement, ensuring more cautious and stable market state predictions. These two approaches are illustrated in Figure 6. Introducing both strategies allows flexibility in the analysis and enables us to test the robustness of the framework more comprehensively under different market circumstances and varying investor mindsets.



(a) Voting Classifier - XGBoost and HMM

(b) Voting Classifier - BaggingClassifier and HMM

Figure 5. Voting classifier architecture combining tree-based models with HMM. (a) XGBoost-HMM; (b) BaggingClassifier-HMM. Each model generates independent predictions, which are then aggregated through a voting mechanism.



Y1	Y2	Y1 ^ Y2
Bull	Bull	Bull
Bear	Bear	Bear
Bull	Neutral	Bull
Bear	Neutral	Bear
Bull	Bear	Neutral

Aggressive Strategy

Y1	Y2	Y1 ^ Y2
Bull	Bull	Bull
Bear	Bear	Bear
Bull	Neutral	Neutral
Bear	Neutral	Neutral
Bull	Bear	Neutral

Conservative Strategy

Figure 6. Voting mechanism combining predictions from tree-based ensemble methods (XGBoost and BaggingClassifier) and HMM.

The rationale behind employing this voting approach is two-fold: The ensemble-based models excel at capturing the short-term, volatile, and nonlinear relationships frequently observed in financial time series (Chen and Guestrin, 2016; Breiman, 2001). Meanwhile, the HMM adeptly captures the market regimes guided by the underlying macroeconomic fundamentals, thus providing a crucial strategic context (Rabiner, 1989; Hamilton, 1989). Integrating these perspectives, the combined voting strategy leverages short-term signals for immediacy and responsiveness alongside stable long-term regime identification from macroeconomic indicators, achieving a comprehensive predictive capability (Ang and Timmermann, 2012; Guidolin and Timmermann, 2007; Nystrup et al., 2017). This combination is particularly effective in financial markets characterized by rapid regime shifts and volatility clustering, as it exploits both granular short-term signals and broader market regimes for decision-making (Krauss et al., 2017; Fischer and Krauss, 2018).

4.9. Trading strategy

The predictive capabilities of the models are leveraged to develop a simple trading strategy, which is assessed using conventional performance metrics, which will be discussed in Section 5. This regime-aware strategy dynamically adjusts its asset allocation according to the predicted market state—bull, bear, or neutral. Specifically, during bull markets, the strategy maintains a fully invested long position in the Russell 3000 or the S&P 500 index. In neutral markets, it shifts its capital to 3-month treasury bills; during bear markets, it adopts a short position on the Russell 3000 or the S&P 500 index. For exposure to the two indices, we have used four ETFs, namely the iShares Russell 3000 index ETF, iShares S&P 500 index ETF, Vanguard 500 index ETF, and the Vanguard Russell 3000 index ETF. We have tested our strategy on all ETFs separately for each of the models in our analysis. The dynamic nature of the strategy and how it rebalances its positions based on the market regime is shown in Figure 7.

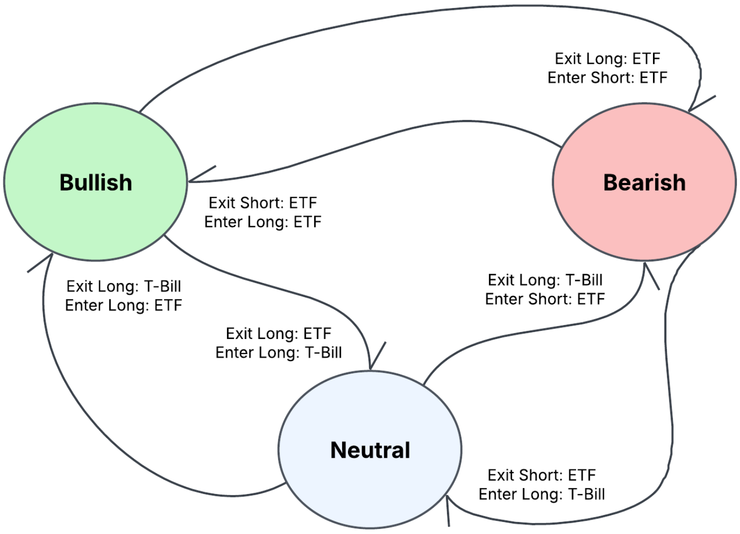
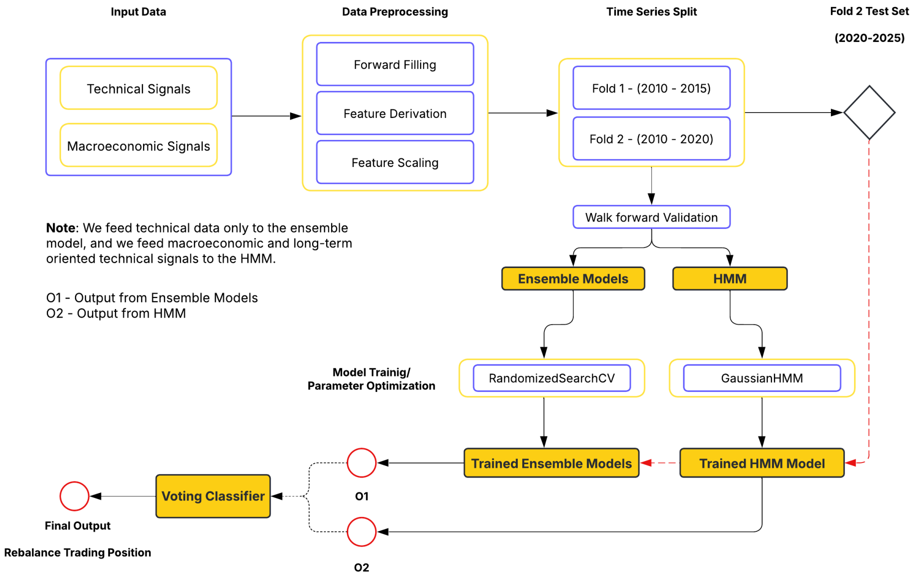


Figure 7. Trading strategy based on market regime shifts.

The performance of this strategy is evaluated against two benchmarks. An upper benchmark is established as the oracle strategy, which assumes perfect foresight of market regimes using oracle labels and employs the same investment approach as our strategy described above. A lower benchmark is defined as the buy-and-hold return over the specified trading period, which simply refers to the strategy of buying the ETF at the start of the trading period and selling it at the end.

4.10. *Model flow chart and summarization*

Figure 8 provides a comprehensive overview of the voting mechanism from the ground up. To recap the flow starting from the data: The complete dataset spanning 2010 to 2025 is used in a walk-forward framework (Figure 2) to ensure a temporally consistent evaluation. Specifically, we apply a time series split with two folds, where each fold preserves the chronological order of the data. In Fold 1, the model is trained on data from 2010 to 2015 and is tested on data from 2015 to 2020. In Fold 2, the training window extends from 2010 to 2020, and testing is performed on the period 2020 to 2025.



Note: We feed technical data only to the ensemble model, and we feed macroeconomic and long-term oriented technical signals to the HMM.

O1 - Output from Ensemble Models
O2 - Output from HMM

Figure 8. Flow chart of the voting classification model.

Two modeling pipelines are developed in parallel. The first pipeline employs ensemble models trained exclusively on technical indicators. Within each fold, the model's hyperparameters are optimized using RandomizedSearchCV (Figure 3) with internal time series cross-validation, ensuring that the model's tuning adheres to the temporal structure of financial data. The second pipeline involves an HMM trained on macroeconomic and long-horizon technical indicators. This model captures latent market regimes and their transitions, thus modeling the underlying structure of financial markets. Like the ensemble models, the HMM is evaluated using the same walk-forward validation approach, implemented with the GaussianHMM available in Python (hmmlearn developers, 2024).

The predictions from both models are then combined using a voting classifier, which integrates signals from both the short-term (ensemble-based) and long-term (HMM-based) perspectives to produce a final decision. The final fold (2020–2025) is treated as the true test window, and the resulting composite signal is used to dynamically rebalance the portfolio positions during this period, as illustrated in Figure 7.

5. Evaluation and results

To evaluate the model's performance, we focus on the effectiveness of the final trading strategy derived from the predicted market regimes. Since our framework integrates both tree-based ensemble methods and the HMM model, the quality of regime classification directly influences the trading decisions. These models collectively generate regime labels that feed into a hybrid voting classifier, enhancing classification stability. The resulting signals inform position entries; therefore, the overall profitability and risk-adjusted returns of the trading strategy serve as practical indicators of the model's efficacy. In this context, a well-performing — according to risk-adjusted metrics and stability — regime-aware strategy reflects the combined predictive power of the underlying ensemble components (Ballings et al., 2015; Henrique et al., 2019).

5.1. Evaluation metrics

To assess the effectiveness of our regime-aware trading strategy, we utilize a comprehensive set of performance metrics capturing both profitability and risk-adjusted returns. These include return-based metrics (e.g., cumulative and annualized return), downside risk measures (e.g., drawdown and ulcer index), and risk-adjusted return ratios (e.g., Sharpe and Sortino ratios). This combination ensures a well-rounded evaluation of not only the profitability of the strategy but also how consistently and efficiently it performs under different market conditions. A detailed description, formulation, and interpretation of these performance metrics can be found in Table 4. These metrics are widely adopted in quantitative finance and trading strategy evaluation (Bailey et al., 2014). We have used the quantstats library (Developers, 2025) to compute all these metrics.

Furthermore, beyond evaluating the performance of the trading strategy itself, we analyze the overall nature of the detected regimes and the differences between the models. We comment on the implicit risk-taking preferences exhibited by different models and discuss why certain models enable better risk-adjusted directional positioning. In addition, we provide a comparative discussion of the trading frequency associated with each method. While some approaches yield marginally higher Sharpe ratios, these results are reported without accounting for the transaction costs. Incorporating even a modest fixed cost per trade could materially alter the effective returns, particularly for models with high trading frequency—or, in other words, higher turnover—thereby reshaping the relative performance rankings.

5.2. Results

The empirical results indicate that the voting classifier combining XGBoost and HMM provides performance that is at least on par with, and in most cases, superior to the other models considered. This improvement is consistently observed across multiple dimensions, including risk-adjusted returns, trading frequency, reduced maximum drawdowns, diminished annualized volatility, and elevated Sharpe and Sortino ratios, as can be noticed in Tables 11, 12, 13, 14, 15, 16, 17, and 18, providing results for all methods over all ETFs, as well as Tables 5 and 6, providing insights into the number of trades by each method for both conservative and aggressive implementation.

Given that our framework incorporates two distinct execution strategies, conservative and aggressive, and is evaluated on two benchmark indices, namely the Russell 3000 and the S&P 500 index, each represented through two ETFs (iShares and Vanguard), the analysis naturally yields eight possible cases for comparison. To ensure clarity, we present the empirical findings for each strategy-index combination separately. We then combine these results to provide a comprehensive perspective, highlighting both commonalities and divergences across the different settings.

Russell 3000 – aggressive: For the aggressive strategy, using both the iShares and Vanguard Russell 3000 ETFs (Tables 11 and 13), the XGBoost–HMM voting framework consistently delivers superior performance relative to the other models. For the iShares ETF, the voting classifier achieves an annualized return of 17.86% with a Sharpe ratio of 1.34, outperforming HMM (12.89%, Sharpe: 0.92), bagging (13.68%, Sharpe: 1.13), and bagging–HMM (16.48%, Sharpe: 1.29). Similarly, for the Vanguard ETF, the framework achieves an annualized return of 18.77% with a Sharpe ratio of 1.47, exceeding HMM (13.75%, Sharpe: 1.01), bagging (13.47%, Sharpe: 1.11), and bagging–HMM (17.10%, Sharpe: 1.41). While standalone XGBoost achieves slightly higher returns (21.98% and 21.74% for iShares and Vanguard, respectively), this comes at the cost of increased annualized risk (20.58% and 20.01%) and lower win rates (56.95% and 57.47%). In contrast, the XGBoost–HMM voting framework balances return with risk more effectively, offering lower volatility (18.46% and 17.43%) and improved win rates of 70.47% and 70.84%.

Another notable advantage of the voting approach in the aggressive setting is its stability, as reflected in both drawdown and trade frequency. The framework achieves reduced maximum drawdowns of −13.40% (iShares) and −10.55% (Vanguard), in comparison with HMM (−30.19%, −30.24%) and XGBoost (−16.45%, −15.70%). Furthermore, as observed in Table 5, the number of trades under the aggressive strategy is 22 for the voting framework compared with 67 for Bagging and 35 for XGBoost.

Russell 3000 – conservative: For the conservative strategy, using Russell 3000 ETFs (Tables 12 and 14), the XGBoost–HMM voting framework continues to demonstrate superior overall performance when evaluated across the return, stability, and efficiency dimensions. For the iShares ETF, the voting classifier with boosting achieves an annualized return of 19.95% and a Sharpe ratio of 1.64, outperforming HMM (12.89%, Sharpe: 0.92), bagging (13.68%, Sharpe: 1.13), and closely matching bagging–HMM (13.49%, Sharpe: 1.70). For the Vanguard ETF, a similar picture emerges: The voting framework achieves an annualized return of 19.67% with a Sharpe ratio of 1.68, well above HMM (13.75%, Sharpe: 1.01) and bagging (13.47%, Sharpe: 1.11), while bagging–HMM, though posting a lower return of 13.52%, records a strong Sharpe ratio of 1.71. These results highlight that bagging–HMM, despite lower raw returns, delivers strong risk-adjusted outcomes by significantly reducing volatility (10.72% and 10.73% for iShares and Vanguard, respectively) and limiting drawdowns (−10.09% and −9.54%), making it an appealing option in terms of downside protection.

From a stability perspective, both voting variants outperform their base models. The boosting–HMM classifier reduces maximum drawdowns to −10.45% (iShares) and −10.55% (Vanguard), improving substantially over HMM (−30.19%, −30.24%) and standalone boosting (−16.45%, −15.70%). The ulcer index values of 2.68 and 2.63 further emphasize smoother recovery compared to boosting (4.61 and 4.50). Bagging–HMM also achieves attractive stability metrics, with drawdowns of −10.09% and −9.54% and ulcer indices of 1.61 and 1.53, the lowest among all methods. While bagging–HMM is especially strong on the stability front, it comes at the cost of lower annualized returns, underscoring the trade-off between absolute and risk-adjusted performance.

S&P 500 – aggressive: For the S&P 500 ETFs under the aggressive strategy (Tables 15 and 17), the performance of the voting framework is weaker than that observed on the Russell 3000 ETFs. While the voting (boosting–HMM) model continues to deliver competitive returns of 16.48% (iShares) and 16.49% (Vanguard) with Sharpe ratios around 1.3, these results represent a notable degradation relative to the Russell 3000, where the same framework achieved returns closer to 18% with Sharpe ratios above 1.6. Risk measures confirm this underperformance: Maximum drawdowns deepen to nearly −19% for the iShares ETF and −18% for the Vanguard ETF, compared with around −13% on the Russell 3000. The framework also generates more trades in this setting, issuing 30 positions compared with 22 in the Russell 3000. The regime allocations provide insight into this behavior: The S&P 500 exhibits a stronger bias toward bullish states, with the voting models allocating more than 75% of the time to bull regimes and less than 20% to neutral phases (Table 9). By contrast, the Russell 3000 displays a more balanced distribution across bull and neutral states. One explanation is the difference in index composition: The S&P 500, being large-cap dominated, tends to move more uniformly with macro trends, which limits regime diversity and amplifies short-term noise in signals, whereas the Russell 3000, covering a broader cross-section of firms, captures greater dispersion across sectors and market capitalizations, providing more nuanced regime transitions.

The voting (boosting–HMM) model clearly improves upon HMM and bagging in both return and risk. Compared with HMM, which produces returns of only 8.4% and Sharpe ratios near 0.6, the voting framework nearly doubles returns while substantially reducing drawdowns from over −31% to below −19%. However, when evaluated against standalone boosting, the performance of the voting classifier is less compelling. Boosting records higher returns, lower volatility (around 15.7% compared with 17.4% for voting), and stronger Sharpe ratios. Although the voting classifier produces marginally smaller drawdowns than boosting, this advantage is not sufficient to compensate for the broader efficiency of the boosting model under a strongly bullish trend.

S&P 500 – conservative: For the S&P 500 ETFs under the conservative strategy (Tables 16 and 18), the relative performance of the voting framework improves compared with its aggressive counterpart. The voting (boosting–HMM) model records annualized returns of 12.83% for the iShares ETF and 12.80% for the Vanguard ETF, somewhat below the aggressive configuration (16.48% and 16.49%), but with markedly reduced volatility (12.88% and 12.86% compared with 17.44% and 17.41%).

When comparing across models within the conservative setting, the advantages of the voting framework are also evident. The boosting–HMM variant substantially outperforms HMM (8.46% and 8.44% returns; Sharpe ratios around 0.63) by both raising returns and reducing drawdowns from over −31% to around −16.8%. Relative to bagging, the voting classifier fails to improve on returns marginally in absolute terms but dominates on stability, with drawdowns of −16.8% compared with bagging's −22.6%. The comparison with boosting is particularly noteworthy: While standalone boosting continues to deliver higher returns (17.7%), it does so at the cost of a much higher volatility (15.7%).

Downside protection further underscores the appeal of the conservative approach. Maximum drawdowns for the voting (boosting–HMM) framework are contained at −16.8%, compared with −22.7% for bagging, −19.1% for boosting, and over −31% for HMM. Moreover, the conservative strategy reduces the number of trades: Only 14 for boosting–HMM and 12 for bagging–HMM (Table 6), compared with 30 and 37 under the aggressive framework. This shift not only implies lower transaction costs but also indicates that the signals generated under the conservative configuration are more stable.

Discussion on trading frequency: We now provide a discussion on the trading frequency observed across different models and its implications. As shown in Tables 5 and 6, our proposed voting-based framework consistently exhibits significantly more stable trading behavior—reflected in markedly fewer trade signals—compared with baseline ensemble models.

For instance, with the Russell 3000 ETFs (Table 5), the boosting method generates 35 trades under both the aggressive and conservative regimes, whereas the voting (boosting–HMM) variant triggers only

22 and 18 trades respectively—a reduction of 37% and 49%. Similarly, bagging results in 67 trades across both regimes, while voting (bagging–HMM) reduces this to 50 (aggressive) and 22 (conservative). A similar trend holds for the S&P 500 ETFs (Table 6): Boosting and bagging produce 27 and 32 trades, respectively, while their voting counterparts reduce these to 14 and 12 in the conservative regime.

This evidence strongly supports the stability of the signal generation in our proposed framework. In real-world trading, each trade incurs a transaction fee. These costs directly reduce the realized returns, thereby lowering the numerator in the Sharpe calculation, while the standard deviation of returns (denominator) remains unaffected. Consequently, a strategy that executes a significantly larger number of trades, without improving the absolute returns, suffers from a degraded risk-adjusted performance.

This phenomenon is clearly observable in our evaluation. While the XGBoost-based model delivers marginally higher returns, it does so at the cost of nearly twice the number of trades in most cases. Although our reported results do not explicitly account for transaction costs, incorporating such real-world frictions would only amplify the advantage of our framework, as discussed above.

Summarized results: Across the four configurations considered—Russell 3000 aggressive, Russell 3000 conservative, S&P 500 aggressive, and S&P 500 conservative—a consistent pattern emerges regarding the effectiveness of the proposed voting mechanism. In particular, the boosting–HMM variant demonstrates the most robust performance profile, providing superior risk-adjusted outcomes compared with all the baseline models in nearly every setting. The only partial exception is the S&P 500–aggressive case, where the dominance of large-cap stocks and a more persistent bullish environment limited the incremental value of the HMM filter. Even in this context, however, the boosting–HMM model outperformed HMM and bagging by a considerable margin, while still containing drawdowns more effectively than boosting.

A secondary but important finding is the consistent performance of the bagging–HMM model, particularly in the Russell 3000–conservative configuration. While its raw returns were lower than those of the boosting–HMM, its superior stability translated into Sharpe ratios that were highly competitive. This highlights that bagging–HMM can be an attractive option for highly risk-averse investors.

Finally, the comparison of aggressive and conservative strategies yields further insights. Across both indices, the conservative configuration consistently improves risk-adjusted performance, even though the absolute returns are somewhat lower. The conservative design reduces volatility, drawdowns, and ulcer index values, while simultaneously increasing win rates and Sortino ratios. This is particularly evident in the Russell 3000 results, where the conservative strategy achieved nearly 20% returns with Sharpe ratios above 1.6, outperforming any individual baseline model. One explanation for this stability advantage lies in regime allocations: Conservative strategies allocate a higher proportion of time to neutral regimes, which reduces the frequency of trades and limits the probability of poor signals.

In summary, the results across both indices and strategies establish the boosting–HMM voting mechanism as a powerful framework for enhancing trading performance. Furthermore, the conservative implementation of the voting framework stands out for its ability to provide stronger downside protection and superior risk-adjusted performance, whereas the aggressive implementation offers a higher-return but a little noisier trade-off. Together, these results demonstrate that the two approaches can satisfy distinct investment objectives: Conservative voting strategies for stability and downside protection, and aggressive variants for investors prioritizing higher absolute returns within a disciplined risk-control framework.

Table 4. Performance evaluation metrics with formulas and interpretations. Here r_t denotes the return on Day t, r_f refers to the risk-free rate, σ_d indicates the downside deviation, N_{win} and N_{total} represent the number of profitable trades and the total number of trades, respectively.

Metric	Formula / Definition	Interpretation
Annualized return (%)	$\left(\prod_{t=1}^{T}(1+r_t)\right)^{\frac{252}{T}} - 1$	Annualized growth rate over T trading days
Cumulative return (%)	$\left(\prod_{t=1}^{T}(1+r_t) - 1\right) \times 100$	Total return over the full evaluation period
Annualized risk (%)	$\sigma = \sqrt{\frac{252}{T-1} \sum_{t=1}^{T}(r_t - \bar{r})^2}$	Standard deviation of returns; overall volatility
Maximum drawdown (%)	$\max_{t\in[1,T]}\left(\frac{\text{Peak}_t - \text{Trough}_t}{\text{Peak}_t}\right)$	Largest peak-to-trough loss in equity curve
Ulcer index	$\sqrt{\frac{1}{T} \sum_{t=1}^{T}(\text{Drawdown}_t)^2}$	Measures the duration and severity of drawdowns
Sharpe ratio	$\frac{\bar{r} - r_f}{\sigma}$	Excess return per unit of total risk (Sharpe (1966))
Sortino ratio	$\frac{\bar{r} - r_f}{\sigma_d}$	Excess return per unit of downside risk (Sortino and Price (1994))
Calmar ratio	$\frac{\text{Annualized Return}}{\text{Max Drawdown}}$	Annual return relative to maximum drawdown
Gain/pain ratio	$\frac{\text{Average Gain}}{\text{Average Loss}}$	Measures average profitability relative to average loss
Profit factor	$\frac{\text{Gross Profit}}{\text{Gross Loss}}$	Ratio of cumulative profits to cumulative losses
Win rate (%)	$\frac{N_{\text{win}}}{N_{\text{total}}} \times 100$	Percentage of trades that are profitable

Table 5. 2020–2025: Number of trades/trade signals—Russell 3000.

Number of trades	Oracle	HMM	Boosting	Bagging	Vote (boosting–HMM)	Vote (bagging–HMM)
Aggressive	7	6	35	67	22	50
Conservative	7	6	35	67	18	22

Table 6. 2020–2025: Number of trades/trade signals—S&P 500.

Number of Trades	Oracle	HMM	Boosting	Bagging	Vote (boosting–HMM)	Vote (bagging–HMM)
Aggressive	7	17	27	32	30	37
Conservative	7	17	27	32	14	12

Table 7. 2020–2025: Regime proportions—Russell 3000 (aggressive).

Strategy	Bull	Neutral	Bear
Oracle	82.14%	7.02%	10.84%
HMM	37.13%	21.14%	41.73%
Boosting	85.73%	1.17%	13.10%
Bagging	92.82%	3.67%	3.51%
Voting (boosting-HMM)	55.54%	31.12%	13.34%
Voting (bagging-HMM)	56.71%	37.83%	5.46%

Table 8. 2020–2025: Regime proportions—Russell 3000 (conservative).

Strategy	Bull	Neutral	Bear
Oracle	82.14%	7.02%	10.84%
HMM	37.13%	21.14%	41.73%
Boosting	85.73%	1.17%	13.10%
Bagging	92.82%	3.67%	3.51%
Voting (boosting-HMM)	36.27%	53.28%	10.45%
Voting (bagging-HMM)	35.49%	62.48%	2.03%

Table 9. 2020–2025: Regime proportions—S&P 500 (aggressive).

Strategy	Bull	Neutral	Bear
Oracle	82.29%	7.02%	10.69%
HMM	68.75%	21.14%	10.11%
Boosting	88.61%	9.83%	1.56%
Bagging	87.89%	1.17%	10.51%
Voting (boosting-HMM)	77.97%	14.95%	7.08%
Voting (bagging-HMM)	75.52%	18.96%	5.52%

Table 10. 2020–2025: Regime proportions—S&P 500 (conservative).

Strategy	Bull	Neutral	Bear
Oracle	82.29%	7.02%	10.69%
HMM	68.75%	21.14%	10.11%
Boosting	88.61%	9.83%	1.56%
Bagging	87.89%	1.17%	10.51%
Voting (boosting-HMM)	63.02%	31.94%	5.04%
Voting (bagging-HMM)	66.81%	23.09%	10.10%

Table 11. Trading strategy results for iShares Russell 3000 index ETF—aggressive, time: 2020–2025.

Metric	Oracle	HMM	Boosting	Bagging	Voting (boosting-HMM)	Voting (bagging-HMM)	Buy-and-hold
Annualized return (%)	25.21	12.89	21.98	13.68	17.86	16.48	11.30
Cumulative return (%)	395.24	137.01	311.11	149.01	221.88	196.04	114.24
Annualized risk (%)	17.65	20.72	20.58	17.24	18.46	17.77	21.69
Maximum drawdown (%)	−11.08	−30.19	−16.45	−24.07	−13.40	−13.40	−35.22
Ulcer index (%)	3.29	15.42	4.61	7.76	3.97	5.06	9.30
Sharpe ratio	1.87	0.92	1.45	1.13	1.34	1.29	0.80
Sortino ratio	2.91	1.44	2.24	1.62	2.07	2.07	1.12
Calmar ratio	2.28	0.43	1.34	0.57	1.33	1.23	0.32
Gain/pain ratio	0.41	0.22	0.32	0.22	0.36	0.39	0.17
Win rate (%)	59.53	63.28	56.95	56.33	70.47	72.97	54.88
Profit factor	1.41	1.22	1.32	1.22	1.36	1.39	1.17

Table 12. Trading strategy results for iShares Russell 3000 index ETF—conservative, time: 2020–2025.

Metric	Oracle	HMM	Boosting	Bagging	Voting (boosting-HMM)	Voting (bagging-HMM)	Buy-and-hold
Annualized return (%)	25.21	12.89	21.98	13.68	19.95	13.49	11.30
Cumulative return (%)	395.24	137.01	311.11	149.01	264.90	145.99	114.24
Annualized risk (%)	17.65	20.72	20.58	17.24	16.35	10.72	21.69
Maximum drawdown (%)	−11.08	−30.19	−16.45	−24.07	−10.45	−10.09	−35.22
Ulcer index (%)	3.29	15.42	4.61	7.76	2.68	1.61	9.30
Sharpe ratio	1.87	0.92	1.45	1.13	1.64	1.70	0.80
Sortino ratio	2.91	1.44	2.24	1.62	2.82	2.58	1.12
Calmar ratio	2.28	0.43	1.34	0.57	1.91	1.34	0.32
Gain/pain ratio	0.41	0.22	0.32	0.22	0.57	0.59	0.17
Win rate (%)	59.53	63.28	56.95	56.33	80.47	84.38	54.88
Profit factor	1.41	1.22	1.32	1.22	1.57	1.59	1.17

Table 13. Trading strategy results for Vanguard Russell 3000 index ETF—aggressive, time: 2020–2025.

Metric	Oracle	HMM	Boosting	Bagging	Voting (boosting-HMM)	Voting (bagging-HMM)	Buy-and-hold
Annualized return (%)	25.26	13.75	21.74	13.47	18.77	17.10	11.29
Cumulative return (%)	396.62	150.17	305.44	145.81	240.14	207.44	114.08
Annualized risk (%)	17.55	19.71	20.01	17.20	17.43	16.67	20.79
Maximum drawdown (%)	−10.80	−30.24	−15.70	−23.92	−10.55	−13.57	−34.61
Ulcer index (%)	3.25	15.32	4.50	7.93	3.88	5.10	9.42
Sharpe ratio	1.88	1.01	1.48	1.11	1.47	1.41	0.82
Sortino ratio	2.94	1.59	2.25	1.60	2.30	2.28	1.15
Calmar ratio	2.34	0.45	1.38	0.56	1.78	1.26	0.33
Gain/pain ratio	0.41	0.23	0.32	0.22	0.39	0.42	0.17
Win rate (%)	60.39	63.64	57.47	56.88	70.84	73.42	55.16
Profit factor	1.41	1.23	1.32	1.22	1.39	1.42	1.17

Table 14. Trading strategy results for Vanguard Russell 3000 index ETF—conservative, time: 2020–2025.

Metric	Oracle	HMM	Boosting	Bagging	Voting (boosting-HMM)	Voting (bagging-HMM)	Buy-and-hold
Annualized return (%)	25.26	13.75	21.74	13.47	19.67	13.52	11.29
Cumulative return (%)	396.62	150.17	305.44	145.81	258.83	146.49	114.08
Annualized risk (%)	17.55	19.71	20.01	17.20	15.71	10.73	20.79
Maximum drawdown (%)	−10.80	−30.24	−15.70	−23.92	−10.55	−9.54	−34.61
Ulcer index (%)	3.25	15.32	4.50	7.93	2.63	1.53	9.42
Sharpe ratio	1.88	1.01	1.48	1.11	1.68	1.71	0.82
Sortino ratio	2.94	1.59	2.25	1.60	2.81	2.58	1.15
Calmar ratio	2.34	0.45	1.38	0.56	1.86	1.42	0.33
Gain/pain ratio	0.41	0.23	0.32	0.22	0.57	0.61	0.17
Win rate (%)	60.39	63.64	57.47	56.88	81.00	84.84	55.16
Profit factor	1.41	1.23	1.32	1.22	1.57	1.61	1.17

Table 15. Trading strategy results for iShares S&P 500 index ETF—aggressive, time: 2020–2025.

Metric	Oracle	HMM	Boosting	Bagging	Voting (boosting-HMM)	Voting (bagging-HMM)	Buy-and-hold
Annualized return (%)	26.46	8.46	17.71	14.56	16.48	14.93	11.76
Cumulative return (%)	431.35	78.19	219.06	163.06	178.43	169.16	120.51
Annualized risk (%)	17.17	20.38	15.77	18.61	17.44	18.53	21.17
Maximum drawdown (%)	−10.63	−30.96	−19.22	−22.69	−18.98	−26.86	−33.90
Ulcer index (%)	3.07	6.90	5.59	6.77	6.55	6.83	8.89
Sharpe ratio	2.00	0.66	1.53	1.12	1.29	1.05	0.84
Sortino ratio	3.13	0.93	2.30	1.62	1.67	1.52	1.18
Calmar ratio	2.49	0.27	0.92	0.64	0.70	0.56	0.35
Gain/pain ratio	0.44	0.15	0.33	0.22	0.23	0.22	0.18
Win rate (%)	60.27	63.86	60.58	55.58	56.21	55.66	55.15
Profit factor	1.44	1.15	1.33	1.22	1.23	1.22	1.18

Table 16. Trading strategy results for iShares S&P 500 index ETF—conservative, time: 2020–2025.

Metric	Oracle	HMM	Boosting	Bagging	Voting (boosting-HMM)	Voting (bagging-HMM)	Buy-and-hold
Annualized return (%)	26.46	8.46	17.71	14.56	12.83	10.34	11.76
Cumulative return (%)	431.35	78.19	219.06	163.06	147.90	96.76	120.51
Annualized risk (%)	17.17	20.38	15.77	18.61	12.88	14.97	21.17
Maximum drawdown (%)	−10.63	−30.96	−19.22	−22.69	−16.81	−20.69	−33.90
Ulcer index (%)	3.07	6.90	5.59	6.77	5.20	6.76	8.89
Sharpe ratio	2.00	0.66	1.53	1.12	1.41	1.05	0.84
Sortino ratio	3.13	0.93	2.30	1.62	1.98	1.24	1.18
Calmar ratio	2.49	0.27	0.92	0.64	0.64	0.40	0.35
Gain/pain ratio	0.44	0.15	0.33	0.22	0.25	0.16	0.18
Win rate (%)	60.27	63.86	60.58	55.58	69.71	64.56	55.15
Profit factor	1.44	1.15	1.33	1.22	1.25	1.16	1.18

Table 17. Trading strategy results for Vanguard S&P 500 index ETF—aggressive, time: 2020–2025.

Metric	Oracle	HMM	Boosting	Bagging	Voting (boosting-HMM)	Voting (bagging-HMM)	Buy-and-hold
Annualized return (%)	26.56	8.44	17.70	14.61	16.49	14.98	11.79
Cumulative return (%)	434.34	78.02	218.89	163.85	178.64	169.95	121.00
Annualized risk (%)	17.07	20.36	15.73	18.50	17.41	18.49	21.14
Maximum drawdown (%)	−10.58	−31.03	−19.13	−22.56	−18.24	−26.92	−33.99
Ulcer index (%)	3.06	6.85	5.57	6.72	6.48	6.78	8.84
Sharpe ratio	2.02	0.66	1.53	1.12	1.30	1.06	0.84
Sortino ratio	3.16	0.93	2.30	1.63	1.68	1.53	1.19
Calmar ratio	2.51	0.27	0.93	0.65	0.70	0.56	0.35
Gain/pain ratio	0.45	0.15	0.33	0.22	0.23	0.22	0.18
Win rate (%)	60.55	64.06	60.91	55.98	56.61	56.06	55.55
Profit factor	1.45	1.15	1.33	1.22	1.23	1.22	1.18

Table 18. Trading strategy results for Vanguard S&P 500 index ETF—conservative, time: 2020–2025.

Metric	Oracle	HMM	**Boosting**	Bagging	Voting (boosting-HMM)	Voting (bagging-HMM)	Buy-and-hold
Annualized return (%)	26.56	8.44	17.70	14.61	12.80	10.32	11.79
Cumulative return (%)	434.34	78.02	218.89	163.85	147.42	96.62	121.00
Annualized risk (%)	17.07	20.36	15.73	18.50	12.86	14.85	21.14
Maximum drawdown (%)	−10.58	−31.03	−19.13	−22.56	−16.67	−20.54	−33.99
Ulcer index (%)	3.06	6.85	5.57	6.72	5.15	6.69	8.84
Sharpe ratio	2.02	0.66	1.53	1.12	1.40	1.05	0.84
Sortino ratio	3.16	0.93	2.30	1.63	1.97	1.24	1.19
Calmar ratio	2.51	0.27	0.93	0.65	0.65	0.41	0.35
Gain/pain ratio	0.45	0.15	0.33	0.22	0.25	0.16	0.18
Win rate (%)	60.55	64.06	60.91	55.98	69.84	64.77	55.55
Profit factor	1.45	1.15	1.33	1.22	1.25	1.16	1.18

6. Discussion

To conclude, our study demonstrates that a voting classification framework combining XGBoost and HMM can offer investors a more favorable risk-adjusted return profile across multiple regimes. By design, this hybrid approach leverages the strengths of both model types—integrating the short-term predictive capabilities and responsiveness of ensemble learning methods with the long-term structural insights of the probabilistic state modeling inherent in HMM. The voting mechanism enables dynamic adaptation to changing market conditions, while the integration ensures smoother transitions between regimes. Consequently, the proposed voting classification framework mitigates the shortcomings of individual models, enhancing both stability and performance consistency across market regimes.

An important observation from the empirical results is the discrepancy in performance between the ETFs following the same index—the iShares and the Vanguard ETF—even though both theoretically track the same underlying benchmark. This divergence reflects tracking error, a well-documented phenomenon in which an ETF's return deviates from that of its benchmark due to factors such as the fund's structure, liquidity, and replication methodology (El-Hassan and Kofman, 2003). The inclusion of both ETFs in our analysis underscores the practical relevance of tracking error and highlights the necessity of evaluating strategies across similar yet non-identical financial instruments.

Moreover, while our proposed framework demonstrates favorable backtested performance on the selected assets, it is important to acknowledge that the current implementation abstracts away real-world frictions such as execution delays, bid–ask spreads, and slippage. Although we provide evidence through the reduced number of trades that our framework inherently mitigates transaction costs, these additional factors can still significantly erode profitability, particularly for high-turnover strategies (Carhart, 1997; Hasbrouck, 2009), which will eventually harm XGBoost-based methods more than our proposed framework, as they inherently trade more. In this study, we restrict our discussion of transaction costs primarily to trading fees, leaving a more comprehensive treatment of market frictions as a direction for future work.

In addition, the model has not yet been extended to alternative asset classes such as commodities or cryptocurrencies, largely because the fundamental data employed in the HMM component is unavailable and suitable substitutes have not been identified. It is worth emphasizing, however, that the primary objective of this study is not profit maximization per se, but rather to demonstrate that a classification-based regime detection methodology can provide a viable and adaptable trading framework. Incorporating cost-aware optimizations and extending the model to a broader set of asset classes remain important directions for future research.

It is important to note that the modeling choices adopted in this study, namely XGBoost, BaggingClassifier, and HMM, do not preclude the use of other approaches that may prove equally effective, or even superior, in similar contexts. The contemporary machine learning landscape offers a wide spectrum of models, each with distinct strengths. For instance, recurrent neural networks (RNNs), particularly their long short-term memory (LSTM) extensions, have demonstrated strong performance in capturing temporal dependencies and long-range nonlinear dynamics in financial time series (Fischer and Krauss, 2018; Bao et al., 2017). Reinforcement learning (RL) also presents a promising alternative: In particular, offline and robust RL frameworks allow agents to learn optimal trading policies directly from historical data by maximizing cumulative reward. Techniques such as Q-learning, policy gradient methods, and actor–critic algorithms enable dynamic adjustment of strategies in response to evolving market conditions, without frequent retraining (Jiang et al., 2017; Gupta et al., 2025).

More recently, advances in large language models (LLMs) have opened new frontiers for incorporating textual and unstructured data into quantitative trading strategies. LLMs have already shown the capacity to extract predictive signals from earnings calls, news articles, and analyst reports, and their integration with structured time-series modeling remains an active area of exploration by both academics and practitioners. Given that leading asset managers such as Vanguard and BlackRock are actively investigating these technologies, the prospect of combining traditional machine learning with language-based approaches is of considerable practical relevance.

On the regime modeling side, while our framework utilizes a multivariate Gaussian HMM for tractability and efficiency, there are several extensions that may offer improved flexibility. Notably, stochastic volatility models replace discrete latent states with continuous ones, allowing volatility dynamics to evolve smoothly over time rather than switching abruptly between regimes (Jacquier et al., 1994; Kim et al., 1998; Clark and Ravazzolo, 2015). Other alternatives include Markov-switching multifractal models and state-space models with non-Gaussian emissions, both of which have been applied with success in asset-pricing contexts (Calvet and Fisher, 2002; Durbin and Koopman, 2012).

In conclusion, our observations and results demonstrate that the proposed voting framework offers a novel and effective approach to regime detection, providing a new perspective on how machine learning and statistical methods can be adapted for financial applications. These observations also underscore that the present framework should not be viewed as exhaustive. Rather, it provides effective instantiation of classification-based regime detection, while acknowledging that advances in deep learning, natural language processing, and stochastic volatility modeling remain fertile areas for future research. Extending this work to incorporate such methods not only aligns with ongoing academic efforts but also reflects active innovation within the financial industry.

Use of AI tools declaration

The authors declare they have not used artificial intelligence (AI) tools in the creation of this article.

Author contributions

All authors contributed equally.

Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable comments, which helped to improve the quality of the manuscript.

Conflict of interest

The authors declare no conflict of interest.

Supplementary Materials

The code and data are available upon request by contacting the corresponding author by email.

References

Abid I, Dhaoui A, Goutte S, et al. (2019) Contagion and bond pricing: The case of the ASEAN region. *Res Int Busi Financ* 47: 371–385. https://doi.org/10.1016/j.ribaf.2018.08.010

Adam T, Ötting M, Michels R (2024) Markov-switching decision trees. *AStA Adv Stat Anal* 108: 461–476. https://doi.org/10.1007/s10182-024-00501-6

AlQaheri H, Panda M (2022) An education process mining framework: Unveiling meaningful information for understanding students' learning behavior and improving teaching quality. *Information* 13: 29. https://doi.org/10.3390/info13010029

Ang A, Bekaert G (2002) International asset allocation with regime shifts. *Rev Financ Stud* 15: 1137–1187. https://www.jstor.org/stable/40302459

Ang A, Timmermann A (2012) Regime changes and financial markets. *Annu Rev Financ Econ* 4: 313–337. https://dx.doi.org/10.2139/ssrn.1919497

Appel G (1979) The moving average convergence-divergence trading method. *Signalert Corp*.

Bailey DH, Borwein JM, De Prado ML, et al. (2014) The probability of backtest overfitting. *J Comput Financ* 20: 39–69. https://dx.doi.org/10.2139/ssrn.2326253

Ballings M, Den Poel DV, Hespeels N, et al. (2015) Evaluating multiple classifiers for stock price direction prediction. *Expert Syst Appl* 42: 7046–7056. https://doi.org/10.1016/j.eswa.2015.05.013

Bao W, Yue J, Rao Y (2017) A deep learning framework for financial time series using stacked autoencoders and long short-term memory. *Neurocomputing* 356: 72–78. https://doi.org/10.1371/journal.pone.0180944

Basak S, Kar S, Saha S, et al. (2019) Predicting the direction of stock market prices using tree-based classifiers. *North Am J Econ Financ* 47: 552–567. https://doi.org/10.1016/j.najef.2018.06.013

Baum LE, Petrie T, Soules G, et al. (1970) A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *Ann Math Stat* 41: 164–171. https://www.jstor.org/stable/2239727

Bergstra J, Bengio Y (2012) Random search for hyper-parameter optimization. *J Mach Learn Res* 13: 281–305. http://jmlr.org/papers/v13/bergstra12a.html

Bollinger JA (2001) *Bollinger on Bollinger Bands*. New York: McGraw-Hill.

Breiman L (2001) Random forests. *Mach Learn* 45: 5–32. https://doi.org/10.1023/A:1010933404324

Bulla J, Bulla I (2006) Stylized facts of financial time series and hidden semi-markov models. *Comput Stat Data Anal* 51: 2192–2209. https://doi.org/10.1016/j.csda.2006.07.021

Calvet LE, Fisher AJ (2002) Multifractality in asset returns: Theory and evidence. *Rev Econ Stat* 84: 381–406. https://www.jstor.org/stable/3211559

Carhart MM (1997) On persistence in mutual fund performance. *J Finance* 52: 57–82. https://doi.org/10.1111/j.1540-6261.1997.tb03808.x

CBOE (2023) Put/call ratios - CBOE. Accessed: 2025-03-29.

Chauvet M, Hamilton JD (2006) Dating business cycle turning points. *Contrib Econ Anal* 243: 1–54. Available from: https://www.nber.org/papers/w11422.

Chen L, Pelger M, Zhu J (2019) Deep learning in asset pricing. *SSRN Electron J*. https://doi.org/10.48550/arXiv.1904.00745

Chen T, Guestrin C (2016) XGBoost: A scalable tree boosting system. In: *Proc 22nd ACM SIGKDD Int Conf Knowl Discov Data Min*, 785–794. https://doi.org/10.1145/2939672.2939785

Christoffersen P, Diebold FX (2006) Financial asset returns, direction-of-change forecasting, and volatility dynamics. *Manage Sci* 52: 1273–1287. Available from: https://www.jstor.org/stable/20110599.

Clark TE, Ravazzolo F (2015) Macroeconomic forecasting performance under alternative specifications of time-varying volatility. *J Appl Econom* 30: 551–575. https://www.jstor.org/stable/26609047

De Prado ML (2018) *Advances in Financial Machine Learning*. Wiley.

Developers Q (2025) QuantStats: Portfolio analytics for quants. Available from: https://github.com/ranaroussi/quantstats.

Dias A, Embrechts P (2010) Modeling exchange rate dependence dynamics at different time horizons. *J Int Money Finance* 29: 1687–1705. https://doi.org/10.1016/j.jimonfin.2010.06.004

Dietterich TG (2000) An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Mach Learn* 40: 139–157. https://doi.org/10.1023/A:1007607513941

Dixon MR, Giroux I, Jacques C, et al. (2018) What characterizes excessive online stock trading? A qualitative study. *J Gambl Issues* 38: 182–203. https://cdspress.ca/?p=8565

Dormann CF, Elith J, Bacher S, et al. (2013) Collinearity: A review of methods to deal with it and a simulation study evaluating their performance. *Ecography* 36: 27–46. https://doi.org/10.1111/j.1600-0587.2012.07348.x

Dorogush AV, Ershov V, Gulin A (2018) CatBoost: Gradient boosting with categorical features support. *arXiv preprint*. https://doi.org/10.48550/arXiv.1810.11363

Durbin J, Koopman SJ (2012) *Time Series Analysis by State Space Methods*. Oxford Univ Press.

El-Hassan N, Kofman P (2003) Tracking error and active portfolio management. *Aust J Manag* 28: 183–207. https://doi.org/10.1177/031289620302800204

Feng G, He J, Polson NG (2018) Deep learning for predicting asset returns. https://doi.org/10.48550/arXiv.1804.09314

Fischer T, Krauss C (2018) Deep learning with long short-term memory networks for financial market predictions. *Eur J Oper Res* 270: 654–669. https://doi.org/10.1016/j.ejor.2017.11.054

Freund Y, Schapire RE (1997) A decision-theoretic generalization of on-line learning and an application to boosting. *J Comput Syst Sci* 55: 119–139. https://doi.org/10.1006/jcss.1997.1504

Friedman JH (2001) Greedy function approximation: A gradient boosting machine. *Ann Stat* 29: 1189–1232. Available from: https://www.jstor.org/stable/2699986.

Guidolin M (2011) Markov switching models in empirical finance. *Adv Econom* 23: 1–86. https://doi.org/10.1108/S0731-9053(2011)000027B004

Guidolin M, Timmermann A (2006) An econometric model of nonlinear dynamics in the joint distribution of stock and bond returns. *J Appl Econom* 21: 1–22. https://dx.doi.org/10.2139/ssrn.582581

Guidolin M, Timmermann A (2007) Asset allocation under multivariate regime switching. *J Econ Dyn Control* 31: 3503–3544. https://doi.org/10.1016/j.jedc.2006.12.004

Gupta R, Pandey A, Pandey A (2025) Can deep reinforcement learning reliably improve dynamic portfolio allocation? *SSRN Electron J.* https://dx.doi.org/10.2139/ssrn.5241923

Hamilton JD (1989) A new approach to the economic analysis of nonstationary time series and the business cycle. *Econometrica* 57: 357–384. https://www.jstor.org/stable/1912559

Hamilton JD (1990) Analysis of time series subject to changes in regime. *J Econom* 45: 39–70. https://doi.org/10.1016/0304-4076(90)90093-9

Hasbrouck J (2009) Trading costs and returns for U.S. equities: Estimating effective costs from daily data. *J Financ* 64: 1445–1477. https://doi.org/10.1111/j.1540-6261.2009.01469.x

Hassan MR, Nath BK (2005) Stock market forecasting using hidden Markov model: A new approach. In: *Proc 5th Int Conf Intell Syst Des Appl (ISDA)* 192–196. IEEE. https://ieeexplore.ieee.org/document/1578783

Hastie T, Tibshirani R, Friedman J (2009) *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2nd ed.

He H, Garcia EA (2009) Learning from imbalanced data. *IEEE Trans Knowl Data Eng* 21: 1263–1284. Available from: https://ieeexplore.ieee.org/document/5128907.

Henrique BM, Sobreiro VA, Kimura H (2019) Literature review: Machine learning techniques applied to financial market prediction. *Expert Syst Appl* 124: 226–251. https://doi.org/10.1016/j.eswa.2019.01.012

hmmlearn developers (2024) hmmlearn: Hidden Markov models in Python. Available from: https://github.com/hmmlearn/hmmlearn.

Hoerl AE, Kennard RW (1970) Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics* 12: 55–67. https://www.jstor.org/stable/1271436

Hull JC (2018) *Options, Futures, and Other Derivatives*. Pearson, 10th ed.

Jacquier E, Polson NG, Rossi PE (1994) Bayesian analysis of stochastic volatility models. *J Bus Econ Stat* 12: 371–389. https://doi.org/10.2307/1392199

Jiang Z, Xu D, Liang J (2017) A deep reinforcement learning framework for the financial portfolio management problem. *arXiv preprint*. https://doi.org/10.48550/arXiv.1706.10059

Khan AA, Chaudhari O, Chandra R (2024) A review of ensemble learning and data augmentation models for class imbalanced problems: Combination, implementation and evaluation. *Expert Syst Appl* 244: 122778. https://doi.org/10.1016/j.eswa.2023.122778

Kim S, Shephard N, Chib S (1998) Stochastic volatility: Likelihood inference and comparison with ARCH models. *Rev Econ Stud* 65: 361–393. https://www.jstor.org/stable/2566931

Krauss C, Do XA, Huck N (2017) Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the S&P 500. *Eur J Oper Res* 259: 689–702. https://doi.org/10.1016/j.ejor.2016.10.031

Kritzman M, Page S, Turkington D (2012) Regime shifts: Implications for dynamic strategies. *Financ Anal J* 68: 22–39. https://doi.org/10.2469/faj.v68.n3.3

Kuhn HW (1955) The Hungarian method for the assignment problem. *Nav Res Logist Q* 2: 83–97. https://doi.org/10.1002/nav.3800020109

Kuncheva LI, Rodríguez JJ (2012) A weighted voting framework for classifier ensembles. *Knowl Inf Syst* 38: 259–275. https://doi.org/10.1007/s10115-012-0586-6

Kutner MH, Nachtsheim CJ, Neter J, et al. (2005) *Applied Linear Statistical Models*. McGraw-Hill/Irwin, New York, 5th ed.

Michels R, Adam T, Ötting M (2023) Tree-based regression within a hidden Markov model framework. *Book of Short Papers - CLADAG2023*.

Morgan J, Reuters (1996) *RiskMetrics™—Technical Document*. Morgan Guaranty Trust Company of New York, New York, 4th ed.

Murphy JJ (1999) *Technical Analysis of the Financial Markets: A Comprehensive Guide to Trading Methods and Applications*. New York Inst Finance.

Neely CJ, Rapach DE, Tu J, et al. (2014) Forecasting the equity risk premium: The role of technical indicators. *Manage Sci* 60: 1772–1791. https://www.jstor.org/stable/42919633

Nguyen N, Nguyen D (2015) Hidden Markov model for stock selection. *Risks* 3: 455–473. https://doi.org/10.3390/risks3040455

Nystrup P, Madsen H, Lindström E (2017) Long memory of financial time series and hidden Markov models with time-varying parameters. *J Forecast* 36: 989–1002. https://doi.org/10.1002/for.2447

Oelschläger L, Adam T, Michels R (2024) fHMM: Hidden Markov models for financial time series in R. *J Stat Softw* 109: 1–25. https://doi.org/10.18637/jss.v109.i09

Pardo L (2006) *Statistical Inference Based on Divergence Measures*, *Monogr Stat Appl Probab*. Chapman and Hall/CRC. https://doi.org/10.1201/9781420034813

Patel J, Shah S, Thakkar P, et al. (2015) Predicting stock market index using fusion of machine learning techniques. *Expert Syst Appl* 42: 2162–2172. https://doi.org/10.1016/j.eswa.2014.10.031

Pedregosa F, Varoquaux G, Gramfort A, et al. (2011) Scikit-learn: Machine learning in Python. *J Mach Learn Res* 12: 2825–2830

Polikar R (2006) Ensemble based systems in decision making. *IEEE Circuits Syst Mag* 6: 21–45. https://ieeexplore.ieee.org/document/1688199

Quinlan JR (1986) Induction of decision trees. In: *Mach Learn* 1: 81–106. Springer. https://doi.org/10.1007/BF00116251

Rabiner LR (1989) A tutorial on hidden Markov models and selected applications in speech recognition. *Proc IEEE* 77: 257–286. https://ieeexplore.ieee.org/document/18626/

Rapach DE, Strauss JK, Zhou G (2010) Out-of-sample equity premium prediction: Combination forecasts and links to the real economy. *Rev Financ Stud* 23: 821–862. https://doi.org/10.1093/rfs/hhp063

Roberts HV (1959) Stock-market "patterns" and financial analysis: Methodological suggestions. *J Finance* 14: 1–10. https://www.jstor.org/stable/2976094

Russell Investments (2020) Market cycles: What they are and why they matter.

Rydén T, Teräsverta T, Åsbrink S (1998) Stylized facts of daily return series and the hidden Markov model. *J Appl Econom* 13: 217–244. https://www.jstor.org/stable/223228

Schapire RE (1990) The strength of weak learnability. *Mach Learn* 5: 197–227. https://doi.org/10.1007/BF00116037

Sharpe WF (1966) Mutual fund performance. *J Bus* 39: 119–138. https://www.jstor.org/stable/2351741

Shi Z, Wu Z, Shi S, et al. (2022) High-frequency forecasting of stock volatility based on model fusion and a feature reconstruction neural network. *Electronics* 11: 4057. https://doi.org/10.3390/electronics11234057

Sortino FA, Price LN (1994) Performance measurement in a downside risk framework. *J Invest* 3: 59–64. Available from: https://www.pm-research.com/content/iijinvest/3/3/59.

Tashman LJ (2000) Out-of-sample tests of forecasting accuracy: An analysis and review. *Int J Forecast* 16: 437–450. https://doi.org/10.1016/S0169-2070(00)00065-0

Tibshirani R (1996) Regression shrinkage and selection via the lasso. *J R Stat Soc Ser B Methodol* 58: 267–288. https://www.jstor.org/stable/2346178

Tsay RS (2010) *Analysis of Financial Time Series*. Wiley, 3rd ed.

Viterbi AJ (1967) Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Trans Inf Theory* 13: 260–269. https://ieeexplore.ieee.org/document/1054010

Wilder JW (1978) *New Concepts in Technical Trading Systems*. Trend Research.

Yardeni Research (2023) Market valuation & bond yields.

Zhang JM, Harman M, Ma L, et al. (2022) Machine learning testing: Survey, landscapes and horizons. *IEEE Trans Softw Eng* 48: 1–36. https://ieeexplore.ieee.org/document/9000651/

Zhou ZH (2012) *Ensemble Methods: Foundations and Algorithms*. Taylor & Francis.