**Yönetim Bilimleri Dergisi/Journal of Administrative Sciences**
Cilt/Volume: 24, Sayı/No: 59, ss./pp.: 271-298
**DOI:** https://doi.org/10.35408/comuybd.1667062

-RESEARCH ARTICLE-

# BAYESIAN-OPTIMIZED ENSEMBLE LEARNING FOR MULTI-CLASS TRADING SIGNAL CLASSIFICATION

Cemal ÖZTÜRK[1]

***Abstract***

*This study tests a practical machine-learning pipeline to predict daily Buy/Hold/Sell trading signals for Apple (AAPL) and to assess whether "good classification" also yields good trading returns after costs. The dataset is built from synchronized daily market series and AAPL-based technical indicators. The target signal is generated by a transparent rule using MACD relative to its signal line and an RSI filter, so the task is a supervised three-class classification problem. Four tree-based ensemble models are compared: Random Forest, LightGBM, XGBoost, and AdaBoost. To avoid fragile, hand-picked settings, each model is tuned with a systematic search procedure. Because the raw labels are strongly imbalanced, SMOTE is applied for training, while all performance and economic tests are run on the original time-ordered test period to keep the evaluation realistic. The results show a clear ranking. XGBoost delivers the best overall classification quality (Accuracy 0.974, Precision 0.975, Recall 0.974, F1 0.974). LightGBM and Random Forest follow at similarly high levels, while AdaBoost is much weaker (Accuracy 0.668, F1 0.536) despite relatively higher precision (0.779), meaning its predictions are not well balanced across classes. Confusion-matrix evidence supports this: the strong models classify Buy and Sell almost perfectly, and most remaining errors come from the Hold class. AdaBoost, however, fails to detect Hold and instead generates many Buy/Sell signals on Hold days. Economic backtests confirm the same story under realistic transaction costs and initial capital. Trading on predicted signals yields +49.1% for XGBoost, +46.1% for LightGBM, and +44.9% for Random Forest. AdaBoost loses money (−11.3%), with worse risk outcomes (Sharpe −0.10, max drawdown 29.0%) and heavier trading (about 68 trades, higher total costs). Overall, modern gradient-boosting ensembles are both statistically strong and economically more credible for this signal design.*

**Keywords:** *Machine Learning, Trading Signal Classification, Technical Indicators, Algorithmic Trading, Bayesian Hyperparameter Optimization.*

**JEL Codes:** *C45, C53, G17, C63, G11.*

**Başvuru:** *27.03.2025* **Kabul:** *18.01.2026*

[1]Dr., Department of Economics, Iğdır University, Iğdır, Türkiye, cemal.ozturk@igdir.edu.tr, https://orcid.org/0000-0003-3850-7416

# BAYESYEN OPTİMİZASYONLU TOPLULUK ÖĞRENMESİ İLE ÇOK SINIFLI ALIM-SATIM SİNYALİ SINIFLANDIRMASI[2]


## *Öz*

*Bu çalışma, Apple (AAPL) için günlük Al/Tut/Sat işlem sinyallerini tahmin eden pratik bir makine öğrenmesi hattını test etmekte ve "iyi sınıflandırma" başarısının, işlem maliyetleri eklendiğinde iyi bir ekonomik performansa dönüşüp dönüşmediğini incelemektedir. Veri seti, senkronize edilmiş günlük piyasa serileri ile AAPL'ye ait teknik göstergelerden oluşturulmaktadır. Hedef sinyal, MACD'nin sinyal çizgisiyle karşılaştırılması ve RSI filtresi kullanan şeffaf bir kuralla üretildiğinden, problem denetimli bir üç sınıflı sınıflandırma problemine dönüşmektedir. Çalışmada dört ağaç tabanlı topluluk modeli karşılaştırılmaktadır: Random Forest, LightGBM, XGBoost ve AdaBoost. Sonuçların ad hoc parametre seçimlerine duyarlı olmaması için her model sistematik bir arama prosedürüyle ayarlanmaktadır. Ham etiketlerde ciddi sınıf dengesizliği bulunduğundan eğitim aşamasında SMOTE uygulanmakta; ancak tüm performans ve ekonomik testler, gerçekçi değerlendirme için orijinal zaman sıralı test döneminde yürütülmektedir. Bulgular, modeller arasında belirgin bir sıralama ortaya koymaktadır. XGBoost en yüksek sınıflandırma kalitesini sunmaktadır (Doğruluk 0.974, Kesinlik 0.975, Duyarlılık 0.974, F1 0.974). LightGBM ve Random Forest çok yakın düzeylerde onu izlemektedir. AdaBoost ise belirgin biçimde daha zayıf kalmaktadır (Doğruluk 0.668, F1 0.536); ayrıca kesinliği nispeten yüksek görünse de (0.779) sınıflar arasında dengeli bir performans sergileyememektedir. Karışıklık matrisi sonuçları bu tabloyu desteklemekte; güçlü modellerin Al ve Sat sınıflarını neredeyse hatasız ayırdığı, kalan hataların büyük ölçüde Tut sınıfında yoğunlaştığı görülmektedir. Buna karşılık AdaBoost'un Tut sınıfını neredeyse hiç yakalayamadığı ve birçok Tut gününü Al/Sat olarak etiketlediği anlaşılmaktadır. Ekonomik geriye dönük test sonuçları da aynı örüntüyü doğrulamaktadır. Gerçekçi işlem maliyetleri ve başlangıç sermayesi altında, model tahminleriyle işlem yapmak XGBoost için +%49.1, LightGBM için +%46.1 ve Random Forest için +%44.9 getiri üretmektedir. AdaBoost ise zarar yazmaktadır (−%11.3) ve daha olumsuz bir risk profili sergilemektedir (Sharpe −0.10, maksimum düşüş %29.0). Ayrıca daha fazla işlem ürettiğinden (yaklaşık 68 işlem) toplam maliyetleri de daha yüksek gerçekleşmektedir. Genel olarak, bu sinyal tasarımı altında modern gradient boosting tabanlı toplulukların hem istatistiksel olarak daha güçlü hem de ekonomik açıdan daha inandırıcı sonuçlar ürettiği değerlendirilmektedir.*



**Anahtar Kelimeler:** *Makine Öğrenmesi, Alım–Satım Sinyali Sınıflandırması, Teknik Göstergeler, Algoritmik Alım–Satım, Bayesyen Hiperparametre Optimizasyonu.*

**JEL Kodları:** *C45, C53, G17, C63, G11.*

"Bu çalışma Araştırma ve Yayın Etiğine uygun olarak hazırlanmıştır."


[2] Genişletilmiş Türkçe Özet, makalenin sonunda yer almaktadır.

## 1. INTRODUCTION

Financial markets stand at the heart of the global economy, playing a critical role in capital allocation, wealth generation, and economic stability. Over the years, these markets have been shaped by rapid technological advancements, globalization, and shifting economic paradigms. The inherent complexity and volatility of financial markets make them a challenging domain for researchers, traders, and policymakers seeking to predict trading signals and devise effective strategies. These challenges are exacerbated by the vast amounts of heterogeneous data generated daily, which often includes financial indicators, economic reports, news sentiment, and even social media trends.

In this setting, forecasting actionable trading signals is difficult because return dynamics are nonlinear, noisy, and regime-dependent, and because predictive relationships can change over time. Traditional methods of technical analysis have therefore been supplemented by machine learning techniques that leverage large historical datasets to identify patterns and generate data-driven predictions. Recent studies show that machine-learning models can outperform traditional statistical approaches in a range of trading scenarios (Saud and Shakya, 2022; Wang et al., 2021; Cheng et al., 2021).

Integrating machine learning into financial analytics marks a meaningful shift in how market information is processed. Unlike classical parametric models, machine learning methods can handle high-dimensional feature spaces and capture complex interactions across variables. Prior research spans supervised learning models trained on labeled price data (Li and Tam, 2018), ensemble learners that improve stability and generalization (Saifan et al., 2020; Gupta and Kumar, 2023), deep learning architectures designed for sequential patterns in time series (Wang and Yan, 2023; Sebastião and Godinho, 2021), and reinforcement learning frameworks that learn trading policies through interaction with the market environment (Cheng et al., 2021). This body of work collectively emphasizes that predictive performance depends not only on model choice but also on feature design, tuning strategy, and evaluation realism.

Building on this literature, the present study develops an empirical benchmarking framework for multi-class trading-signal prediction with a clear emphasis on practical evaluation and interpretability. We focus on supervised, tree-based ensemble methods—Random Forest, LightGBM, XGBoost, and AdaBoost—because they provide strong predictive baselines while remaining relatively transparent and deployment-friendly in applied settings.

Empirically, we construct a cross-asset dataset for Apple Inc. (AAPL) using the yfinance interface (Aroussi, 2024) and enrich the dataset with standard technical indicators that summarize trend, momentum, and volatility. Trading signals are defined as a three-class target (Buy/Hold/Sell) using a transparent, rule-based labeling scheme derived from MACD and RSI. Because class imbalance is a common feature of signal labels, we apply SMOTE to balance the training set and reduce biased learning toward dominant classes (Chawla et al., 2002). Model hyperparameters are tuned via Bayesian optimization with Optuna (Akiba et al., 2019), and model behavior is interpreted using SHAP-based feature attribution to improve transparency (Lundberg and Lee, 2017). Finally, beyond standard classification metrics, we evaluate whether predictive gains translate into economic relevance using a cost-inclusive backtesting design that accounts for realistic transaction frictions.

Overall, the study contributes a reproducible end-to-end pipeline that links data construction, feature engineering, signal labeling, imbalance handling, model tuning, interpretability, and economic evaluation within a single framework. The remainder of the paper is organized as follows: Section 2 describes the data, signal definition, and modeling workflow; Section 3 reports empirical results and diagnostic evidence; and Section 4 discusses implications and limitations for practical trading applications.

## 2. MATERIALS AND METHODS

### 2.1. Empirical Workflow and Study Pipeline

This section summarizes the empirical workflow used to develop and validate multi-class machine learning models for trading signal prediction. As a compact roadmap, Figure 1 links each stage of the pipeline—from data acquisition and feature construction to economic validation and interpretability—thereby supporting transparency and reproducibility.

As shown in Figure 1, the pipeline proceeds sequentially through data collection (Phase 1), feature engineering (Phase 2), and rule-based signal generation (Phase 3) that defines the Buy/Hold/Sell target space. To mitigate label imbalance prior to model fitting, the workflow incorporates SMOTE-based resampling (Chawla et al., 2002). Competing ensemble learners—Random Forest, LightGBM, XGBoost, and AdaBoost—are then trained (Phase 4), with hyperparameters tuned via Optuna (Akiba et al., 2019) (Phase 5). Model performance is assessed using standard classification metrics together with trading simulation (backtesting) to evaluate decision usefulness under realistic execution logic (Phase 6). Finally, SHAP-based feature attribution and risk-oriented portfolio diagnostics are used to interpret model behavior and quantify economic relevance (Lundberg and Lee, 2017) (Phase 7).

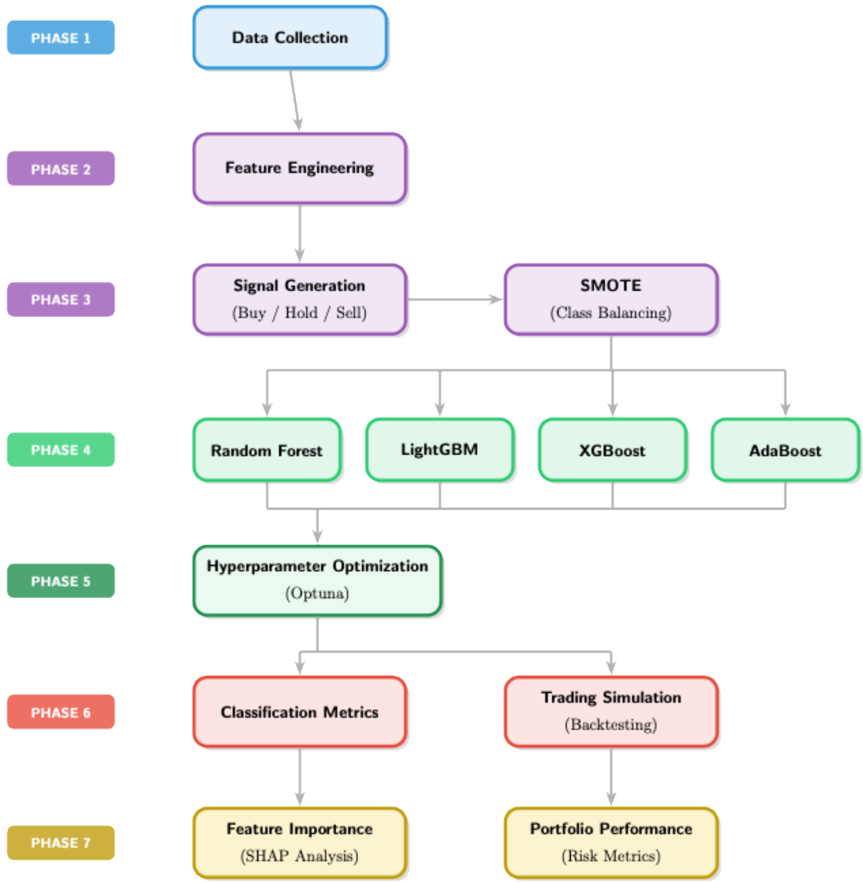


**Figure 1. The Methodological Framework of the Study**

## 2.2. Data and Study Design

### *2.2.1 Data Sources and Asset Universe*

The dataset utilized in this study comprises a broad set of market variables collected via the *yfinance* Python package (Aroussi, 2024). Specifically, we retrieve daily closing prices for the target asset (AAPL) together with a cross-asset indicator set spanning major U.S. equity indices (e.g., NASDAQ, S&P 500, Dow Jones, NYSE), international equity benchmarks (e.g., FTSE 100, Russell 2000), key exchange rates (EUR/USD and GBP/USD), commodities (crude oil and gold), market uncertainty and currency proxies (VIX and U.S. dollar index), selected mega-cap technology stocks, Bitcoin, and the U.S. 10-year Treasury yield. This diversified asset universe is intended to capture common drivers of equity dynamics and to provide a richer information set for learning trading signals.

The empirical dataset is constructed in three sequential steps to ensure a clean, synchronized, and time-consistent sample for predictive modeling. First, the raw daily series are downloaded from Yahoo Finance over the broad collection window 2014-01-02 to 2024-12-05. Second, all instruments are merged into a single panel and aligned by trading date, yielding an initial matrix of 3916 observations. Third, to avoid implicit imputation and ensure a fully observed feature vector at each timestamp, rows with missing values are removed. This step eliminates 1397 rows, resulting in a final modeling sample spanning 2014-09-17 to 2024-12-04, comprising 2519 trading days and 21 input features.

### *2.2.2 Feature Engineering: Technical Indicators*

Building on this synchronized base panel, additional engineered predictors are derived to summarize trends, volatility, and momentum (e.g., moving averages, Bollinger Bands, and related technical indicators) (Lin et al., 2021). In line with standard technical-analysis constructions, the engineered set includes moving averages and Bollinger Band components (Bollinger, 2002), MACD and its signal line (Appel, 1979), and RSI (Wilder, 1978). With these additions, the final predictor set used for model estimation contains 32 input features in total (21 cross-asset close-price features plus 11 AAPL-derived technical-indicator features), excluding the target variable Signal. Recent evidence also suggests that the informativeness of such indicator-based feature spaces can improve when feature quality and selection are explicitly addressed (Ji et al., 2022).

Figure 2 provides an interpretable snapshot of how the engineered technical indicators summarize different dimensions of AAPL's recent market behavior (last 500 trading days of the synchronized sample). Panel (a) overlays the price path with MA7 and MA21, which smooth high-frequency noise and make trend direction easier to diagnose at two horizons. The Bollinger Bands extend this view by forming a volatility envelope around the moving average: episodes where the price persistently leans toward the upper band typically coincide with strong trend continuation and elevated dispersion, whereas compressions and frequent touches of the lower band are consistent with weaker momentum and drawdown phases (Bollinger, 2002).

Panel (b) reports MACD, its signal line, and the corresponding histogram (computed as MACD minus the signal line in the plotting routine). The histogram's sign changes offer a compact way to visualize shifts in medium-term momentum, while large positive/negative swings reflect periods when the fast and slow exponential moving averages diverge materially—often aligning with trend accelerations or reversals (Appel, 1979).

Panel (c) shows the 14-day RSI together with conventional threshold bands at 30 and 70. Rather than treating these cutoffs as deterministic "buy/sell" triggers, the figure is used here as a diagnostic: clustering near the upper region signals sustained buying pressure, whereas dips below 30 indicate stress regimes where mean-reversion dynamics may become more plausible (Wilder, 1978).

Overall, Figure 2 motivates the feature-engineering design adopted in the predictive pipeline: each indicator targets a distinct market attribute—trend (moving averages), volatility state (Bollinger Bands), momentum timing (MACD), and oscillator-based pressure (RSI). This modular structure aligns with recent evidence that feature quality and careful selection of technical indicators can materially affect stock-movement classification performance (Ji et al., 2022).

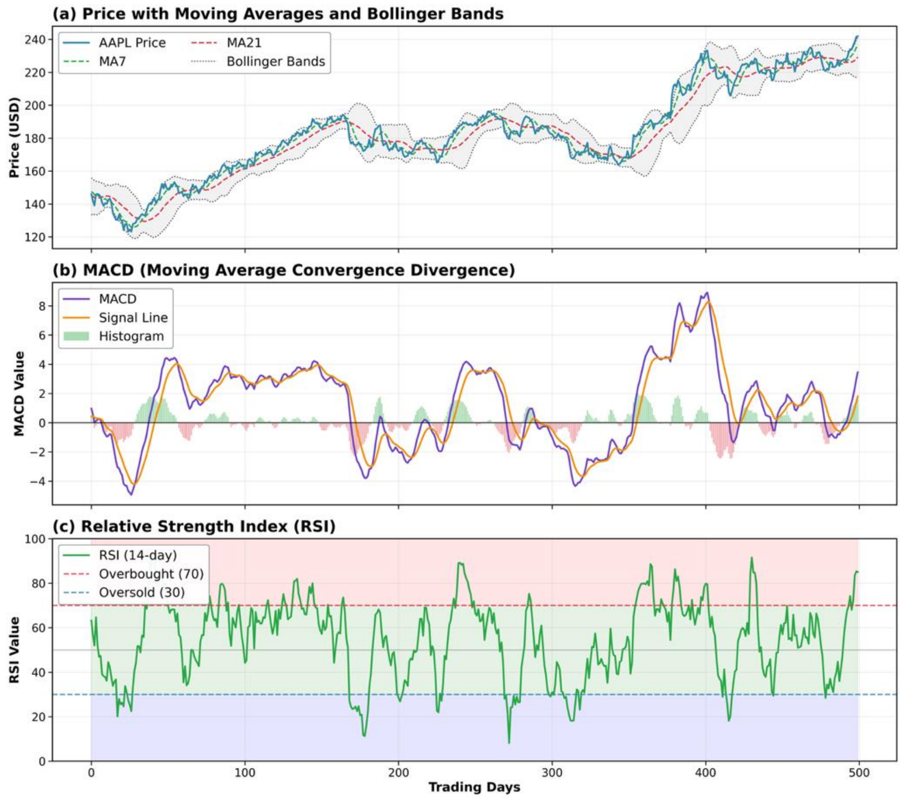


**Figure 2. Technical Indicators for AAPL Based on the Last 500 Trading Days**

Table 1 provides a consolidated description of the full set of variables used in the empirical analysis, covering both the cross-asset close-price predictors and the engineered AAPL-based technical indicators, along with their definitions, units, sampling frequency, and data provenance.

**Table 1. Features and Their Descriptions.**

| Feature Name | Description | Units | Frequency | Source |
|---|---|---|---|---|
| Apple Stock Prices (AAPL) | Daily closing prices of Apple stock, representing the company's equity performance. | USD | Daily | Yahoo Finance (AAPL) |
| NASDAQ Index | NASDAQ Composite Index values, reflecting the performance of tech-heavy equities. | Index Value | Daily | Yahoo Finance (NASDAQ) |
| NYSE Index | NYSE Composite Index values, summarizing the performance of stocks listed on NYSE. | Index Value | Daily | Yahoo Finance (NYSE) |
| S&P 500 Index | S&P 500 Index values, indicating the performance of 500 large-cap US stocks. | Index Value | Daily | Yahoo Finance (S&P 500) |
| Dow Jones Industrial Average | Dow Jones Industrial Average values, representing 30 major US companies. | Index Value | Daily | Yahoo Finance (Dow Jones) |
| EUR/USD Exchange Rate | Daily EUR/USD exchange rates, capturing the relationship between Euro and USD. | EUR/USD | Daily | Yahoo Finance (EUR/USD) |
| GBP/USD Exchange Rate | Daily GBP/USD exchange rates, capturing the relationship between Pound and USD. | GBP/USD | Daily | Yahoo Finance (GBP/USD) |
| FTSE 100 Index | FTSE 100 Index values, representing the 100 largest UK-listed companies. | Index Value | Daily | Yahoo Finance (FTSE 100) |
| RUSSELL 2000 Index | RUSSELL 2000 Index values, representing US small-cap stocks. | Index Value | Daily | Yahoo Finance (RUSSELL 2000) |
| Crude Oil Prices | Daily closing prices of WTI crude oil futures (CL=F), a key global energy cost indicator. | USD/Barrel | Daily | Yahoo Finance (CL=F) |
| Gold Prices | Daily closing prices of gold, reflecting market demand for safe-haven assets. | USD/Ounce | Daily | Yahoo Finance (Gold) |
| Volatility Index (VIX) | Volatility Index (VIX), measuring market uncertainty and risk sentiment. | Index Value | Daily | Yahoo Finance (VIX) |
| USD Index | USD Index, tracking the value of USD against a basket of major currencies. | Index Value | Daily | Yahoo Finance (USD Index) |
| Amazon Stock Prices | Daily closing prices of Amazon stock, reflecting the company's equity performance. | USD | Daily | Yahoo Finance (Amazon) |
| Google Stock Prices | Daily closing prices of Google stock, representing the company's equity performance. | USD | Daily | Yahoo Finance (Google) |
| Microsoft Stock Prices | Daily closing prices of Microsoft stock, reflecting the company's equity performance. | USD | Daily | Yahoo Finance (Microsoft) |
| Nvidia Stock Prices | Daily closing prices of Nvidia stock, representing the company's equity performance. | USD | Daily | Yahoo Finance (Nvidia) |

| | | | | |
|---|---|---|---|---|
| Meta Stock Prices | Daily closing prices of Meta stock, reflecting the company's equity performance. | USD | Daily | Yahoo Finance (Meta) |
| Tesla Stock Prices | Daily closing prices of Tesla stock, representing the company's equity performance. | USD | Daily | Yahoo Finance (Tesla) |
| Bitcoin Prices | Daily Bitcoin prices, capturing the cryptocurrency's market value. | USD | Daily | Yahoo Finance (Bitcoin) |
| US 10-Year Treasury Yield | Daily US 10-Year Treasury Yield, a benchmark for risk-free rates. | Percent | Daily | Yahoo Finance (US 10-Year Treasury Yield) |
| 7-Day Moving Average (MA7) | 7-day moving average of Apple stock prices, indicating short-term trends. | USD | Daily | Derived |
| 21-Day Moving Average (MA21) | 21-day moving average of Apple stock prices, indicating longer-term trends. | USD | Daily | Derived |
| MACD | Difference between the 12-day and 26-day exponential moving averages. | USD | Daily | Derived |
| Signal Line | Signal line of MACD, computed as the 9-day exponential moving average of MACD and used to form the MACD histogram (MACD − Signal_Line) | USD | Daily | Derived |
| 20-Day Standard Deviation (20SD) | Standard deviation of Apple stock prices over 20 days. | USD | Daily | Derived |
| Upper Bollinger Band (upper_band) | Upper Bollinger Band, calculated as MA21 + 2*(20SD). | USD | Daily | Derived |
| Lower Bollinger Band (lower_band) | Lower Bollinger Band, calculated as MA21 - 2*(20SD). | USD | Daily | Derived |
| Exponential Moving Average (EMA) | Exponential moving average of Apple stock prices. | USD | Daily | Derived |
| Relative Strength Index (RSI) | RSI, a momentum indicator measuring overbought/oversold conditions. | Dimensionless | Daily | Derived |
| Momentum | Momentum of Apple stock prices calculated over 4 days. | USD | Daily | Derived |
| Volatility | Volatility of Apple stock prices based on 14-day rolling standard deviation. | Dimensionless | Daily | Derived |
| Signal | Target variable: trading signal (1=Buy, 2=Sell, 0=Hold) based on RSI–MACD rule configuration. | Categories | Daily | Derived |

### *2.2.3 Target Variable: Trading-Signal Definition*

The dependent variable in this study is a rule-based, three-class trading signal constructed from two standard momentum diagnostics: the Moving Average Convergence Divergence (MACD) and the Relative Strength Index (RSI). The intention is to translate the indicator configuration observed at each trading day into an actionable label that can be learned by supervised classifiers. Let $t \in \{1, \dots, T\}$ denote the trading-day index, and let $\mathrm{MACD}_t$, $\mathrm{SL}_t$ (the MACD signal line), and $\mathrm{RSI}_t$ denote the corresponding indicator values available at day $t$. We then define the target label $y_t \in \{0,1,2\}$, where 0 represents Hold, 1 represents Buy, and 2 represents Sell. The labeling rule is specified as the following piecewise mapping:

$$y_t = \begin{cases} 1, & \text{if } \mathrm{MACD}_t > \mathrm{SL}_t \text{ and } \mathrm{RSI}_t < 50, \\ 2, & \text{if } \mathrm{MACD}_t < \mathrm{SL}_t \text{ and } \mathrm{RSI}_t > 50, \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

Intuitively, a Buy label ($y_t = 1$) is assigned when MACD crosses above its signal line—an indicator of upward momentum—while the RSI remains below the neutral threshold of 50, which acts as a conservative filter against entering positions after an already-extended run-up. Conversely, a Sell label ($y_t = 2$) is assigned when MACD falls below the signal line—suggesting downward momentum—while RSI is above 50, reflecting comparatively stronger recent price strength and helping to avoid mechanically selling into uniformly weak conditions. All remaining configurations are labeled Hold ($y_t = 0$), ensuring that "action" classes are reserved for states where the momentum direction and the RSI filter jointly support a clearer trading interpretation.

This construction converts the empirical problem into a three-class classification setting: given the engineered predictor vector $\mathbf{x}_t$, the models are trained to approximate the mapping $\mathbf{x}_t \mapsto y_t$. Because the label $y_t$ is computed from indicator values available at time $t$, it is naturally compatible with the chronological out-of-sample evaluation protocol adopted in the subsequent sections.

*2.2.4 Class Imbalance Handling*

A key practical challenge in the present setting is that the rule-based trading labels are highly imbalanced. In financial classification problems, such an imbalance is not merely a statistical inconvenience; it can meaningfully distort what a model "learns," because many algorithms tend to prioritize the most frequent class unless corrective measures are taken. To mitigate this risk, the study adopts the Synthetic Minority Oversampling Technique (SMOTE), which increases minority-class representation by creating synthetic observations in feature space rather than simply duplicating existing cases (Chawla et al., 2002).

After feature engineering and label construction, the original signal distribution is strongly skewed toward Hold: the sample contains 1934 Hold observations (77.4%), while Buy and Sell appear 148 (5.9%) and 417 (16.7%) times, respectively. This pattern is consistent with realistic trading environments where "no clear action" states are far more common than decisive entry or exit conditions. However, if left unaddressed, such a distribution can lead to models that look accurate on paper but are systematically weak at detecting the rarer Buy and Sell states that are of primary interest.

Formally, SMOTE generates a synthetic instance for a minority-class observation $\mathbf{x}$ by selecting one of its $k$-nearest minority neighbors $\mathbf{x}_{\mathrm{nn}}$ and interpolating along the segment connecting them:

$$\mathbf{x}_{\mathrm{syn}} = \mathbf{x} + \lambda(\mathbf{x}_{\mathrm{nn}} - \mathbf{x}), \qquad \lambda \in (0,1). \tag{2}$$

This mechanism increases minority-class density without duplicating points and typically improves the learner's ability to recover minority-class structure.

In implementation, SMOTE is applied to the feature matrix $\mathbf{X}$ and label vector $\mathbf{y}$ constructed after preprocessing and feature engineering, for which the effective sample size is 2499 observations. After resampling, the dataset expands to 5802 observations and becomes perfectly balanced by design: Hold = 1934, Buy = 1934, and Sell = 1934 (each 33.3%). The resampling procedure is conducted using the *imbalanced-learn* library, which provides a standardized and reproducible SMOTE implementation for machine-learning workflows (Lemaître et al., 2017).

This balancing step is important for two reasons. First, it prevents the training process from being dominated by the Hold class, which would otherwise encourage trivial "always-hold" behavior. Second, it allows model comparisons based on class-sensitive metrics such as Recall and F1-score to reflect genuine discriminative ability rather than class prevalence effects. SMOTE is applied exclusively to the training partition to preserve the integrity of out-of-sample evaluation. For these reasons, SMOTE is treated as a core preprocessing component of the predictive pipeline.

### *2.2.5 Train–Test Split Protocol*

Figure 3 visualizes the daily closing price of Apple Inc. (AAPL) over the finalized sample period 2014-09-17 to 2024-12-04 after the preprocessing and synchronization steps described above. The blue line shows the observed AAPL closing price, while the red dashed vertical line marks the chronological train–test split date (2021-11-05), with approximately 70% of the time-ordered observations allocated to model training and the remaining 30% to out-of-sample testing. Accordingly, the training window covers 2014-09-17–2021-11-05, and the testing window covers 2021-11-05–2024-12-04.

Beyond documenting the evaluation design, the figure provides a compact summary of the price range and variability in the sample: the minimum closing price is $20.60 (2016-05-12) and the maximum is $241.92 (2024-12-04). The overall trajectory exhibits pronounced nonlinearity and volatility clustering, underscoring the relevance of nonparametric and ensemble-based learners. Most importantly, the time-ordered split prevents look-ahead bias and mirrors realistic deployment settings in which models are fitted on historical information and assessed on genuinely unseen future observations; therefore, performance measured on the testing window serves as a more credible indicator of generalization in practical trading scenarios.


**Figure 3. Daily Closing Price of Apple Inc. (AAPL) with Train–Test Split**

## 2.3. Methods

This study formulates trading-signal prediction as a supervised three-class classification problem, where the engineered cross-asset and technical-indicator features are used to predict the target label $y_t \in \{0,1,2\}$ (Hold/Buy/Sell). The modeling stage benchmarks four tree-based ensemble learners—AdaBoost, Random Forest, XGBoost, and LightGBM—which are well-suited to financial data due to their ability to capture nonlinearities, interaction effects, and threshold-type behavior under noise. Class imbalance is explicitly handled by SMOTE in the preprocessing pipeline (Section 2.1.4), while the learners described below focus on discrimination and generalization. For completeness, the methods are presented in approximate historical order together with their mathematical foundations.

### *2.3.1 Adaptive Boosting (AdaBoost)*

Adaptive Boosting (AdaBoost) was introduced by Freund and Schapire (1997) as a procedure that converts a sequence of weak learners into a strong classifier by iteratively reweighting training observations. Intuitively, observations that are misclassified in earlier rounds receive higher weights in subsequent rounds, forcing the algorithm to concentrate on "hard" cases.

In the binary setting, the weight assigned to the $t$-th weak learner $h_t(x)$ is commonly expressed as:

$$\alpha_t = \frac{1}{2} \ln \left( \frac{1 - \varepsilon_t}{\varepsilon_t} \right), \tag{3}$$

where $\varepsilon_t$ is the weighted error rate of $h_t(x)$. The resulting classifier aggregates weak learners through a weighted vote:

$$H(x) = \operatorname{sign}\left(\sum_{t=1}^{T} \alpha_t \, h_t(x)\right). \tag{4}$$

In this study, AdaBoost is implemented in the multi-class setting consistent with the three-class Signal definition; therefore, the final class prediction is obtained via the standard multi-class extension (rather than a binary sign rule), while the underlying principle—iterative reweighting and weighted aggregation—remains the same (Freund and Schapire, 1997). Importantly, potential imbalance in trading labels is not left to AdaBoost alone; it is addressed explicitly via SMOTE as part of the preprocessing workflow.

*2.3.2 Random Forest*

Random Forest, proposed by Breiman (2001), constructs an ensemble of decision trees using bootstrap resampling and random feature subsampling. Each tree is trained on a bootstrap sample of the data, and at each node split a randomly selected subset of predictors is considered. This procedure reduces correlation among trees and typically lowers variance relative to a single decision tree.

For multi-class classification, node splitting is commonly driven by an impurity criterion such as the Gini index:

$$G = \sum_{i=1}^{C} p_i \,(1 - p_i) = 1 - \sum_{i=1}^{C} p_i^2, \tag{5}$$

where $p_i$ denotes the proportion of observations belonging to class $i$ within the node and $C$ is the number of classes. Final predictions are obtained via majority voting across trees, which provides robustness under noisy predictors and improves out-of-sample stability (Breiman, 2001).

*2.3.3 Gradient Boosting Machines (GBM)*

Gradient Boosting Machines (GBM) were popularized by Friedman (2001) as an additive modeling framework in which base learners are fitted sequentially to minimize a specified loss function $L(y, F(x))$. At iteration $t$, the method constructs pseudo-residuals (negative gradients) evaluated at the current model $F_{t-1}$:

$$r_i^{(t)} = -\frac{\partial L\big(y_i, F(x_i)\big)}{\partial F(x_i)} \,\Big|_{F=F_{t-1}}. \tag{6}$$

A weak learner $h_t(x)$ (typically a shallow decision tree) is then trained to approximate these pseudo-residuals, and the model is updated as:

$$F_t(x) = F_{t-1}(x) + \eta\, h_t(x), \tag{7}$$

Where $\eta \in (0,1]$ is the learning rate controlling the step size and, consequently, the bias–variance trade-off (Friedman, 2001). This sequential correction mechanism is a key reason why boosting methods often perform well in complex prediction tasks with nonlinear decision boundaries.

*2.3.4 XGBoost (Extreme Gradient Boosting)*

XGBoost (Chen and Guestrin, 2016) is a scalable and regularized gradient boosting framework that enhances generalization by explicitly penalizing tree complexity. Let the prediction be $\hat{y}_i = \sum_{k=1}^{K} f_k\,(x_i)$, where each $f_k$ is a decision tree. XGBoost minimizes an objective of the form:

$$\mathcal{L}(\theta) = \sum_{i=1}^{N} L(\,y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(\,f_k), \tag{8}$$

where $\Omega(\cdot)$ is a regularization term that discourages overly complex trees. A commonly used form is:

$$\mathcal{L}\Omega(f) = \gamma T + \frac{1}{2}\lambda \parallel \omega \parallel^2, \tag{9}$$

with $T$ denoting the number of leaves and $\omega$ the vector of leaf weights. To make optimization efficient, XGBoost employs a second-order Taylor approximation of the loss around the current prediction, yielding:

$$\mathcal{L}^{(t)} \approx \sum_{i=1}^{N} \left[g_i f(x_i) + \frac{1}{2} h_i f(x_i)^2\right] + \Omega(f), \tag{10}$$

where $g_i$ and $h_i$ are the first and second derivatives of the loss with respect to the current prediction (Chen and Guestrin, 2016). This second-order structure supports more precise split decisions and typically improves performance under complex nonlinear patterns.

*2.3.5 LightGBM*

LightGBM is a gradient-boosted decision tree framework designed for computational efficiency and scalability (Ke et al., 2017). While it follows the same additive boosting principle as GBM, it accelerates training through techniques such as histogram-based binning and efficient split finding, which is advantageous when repeated hyperparameter optimization is conducted and when the feature space is relatively high-dimensional. In the present context, these design choices make LightGBM a practical benchmark alongside XGBoost for multi-class trading-signal classification.

Overall, the study combines AdaBoost and Random Forest as established ensemble baselines with two modern gradient-boosting implementations (XGBoost and LightGBM). Together, these methods provide a coherent modeling set for learning nonlinear mappings from cross-asset and technical-indicator features to multi-class trading signals under realistic market noise and imbalanced label structures.

### 2.4. Model Evaluation and Backtesting

Model performance is evaluated along two dimensions to capture both predictive accuracy and economic relevance. First, multi-class classification quality is assessed on the chronological out-of-sample test window using confusion-matrix-based metrics. Second, predicted labels are converted into trades and evaluated via a cost-inclusive trading simulation that reflects realistic execution frictions.

*2.4.1 Classification Metrics*

Let $y_t \in \{0,1,2\}$ denote the realized signal (Hold/Buy/Sell) and $\hat{y}_t$ the predicted label. Using the standard confusion matrix components $TP$, $TN$, $FP$, and $FN$, we report:

$$\begin{aligned} \text{Accuracy} &= \frac{\text{TP}+\text{TN}}{\text{TP}+\text{TN}+\text{FP}+\text{FN}}, \\ \text{Precision} &= \frac{\text{TP}}{\text{TP}+\text{FP}}, \\ \text{Recall} &= \frac{\text{TP}}{\text{TP}+\text{FN}}, \\ \text{F1-score} &= 2\cdot\frac{\text{Precision}\cdot\text{Recall}}{\text{Precision}+\text{Recall}}. \end{aligned} \tag{11}$$

These metrics summarize overall correctness (Accuracy) and class-sensitive performance (Precision/Recall/F1), which is particularly relevant when the labels are unevenly distributed.

*2.4.2 Cost-Inclusive Backtesting*

Economic performance is evaluated on the original, time-ordered test sample by running a trading simulation driven by model predictions $\hat{y}_t$ (not SMOTE-resampled sequences). The simulator follows a long-only, all-in/all-out rule: it enters a position when $\hat{y}_t = 1$ (Buy) and exits when $\hat{y}_t = 2$ (Sell); $\hat{y}_t = 0$ (Hold) triggers no action. Trading frictions are modeled as a proportional one-way cost rate

$$c = c_{\text{comm}} + c_{\text{slip}} + c_{\text{spr}}, \tag{12}$$

covering commission, slippage, and bid–ask spread. With price $P_t$, cash $C_t$, and holdings $q_t$, portfolio value is tracked as

$$V_t = C_t + q_t P_t. \tag{13}$$

On entry, the effective purchase price is $P_t(1 + c)$ , and the number of shares is set by available cash:

$$q_t = \left\lfloor \frac{C_{t-1}}{P_t(1 + c)} \right\rfloor. \tag{14}$$

On exit, proceeds are reduced proportionally by the same cost rate. From the simulated $\{V_t\}$, we report total return, buy-and-hold return over the same test window, and excess return. Risk and stability are summarized using annualized volatility, Sharpe ratio (when defined), and maximum drawdown, alongside basic trading statistics such as the number of trades and win rate.

## 3. RESULTS AND DISCUSSION

The results provide clear evidence on the comparative performance of the four ensemble learners—Random Forest, LightGBM, XGBoost, and AdaBoost—in multi-class trading-signal prediction. To improve robustness and reduce sensitivity to ad hoc parameter choices, the study adopts a principled hyperparameter optimization procedure implemented via Optuna (Akiba et al., 2019). This approach systematically explores the hyperparameter space and identifies configurations that balance model complexity and predictive stability.

Figure 4 compares the four ensemble classifiers (Random Forest, LightGBM, XGBoost, and AdaBoost) using Accuracy, Precision, Recall, and F1 Score. The results indicate that XGBoost delivers the best overall performance (Accuracy = 0.974; Precision = 0.975; Recall = 0.974; F1 = 0.974), with LightGBM and Random Forest closely trailing at similarly high levels. In contrast, AdaBoost exhibits substantially weaker performance (Accuracy = 0.668; F1 = 0.536), despite a relatively higher precision (0.779). This divergence suggests that AdaBoost's predictions are less balanced, with reduced coverage of the relevant class assignments reflected in its lower recall and F1.

Overall, Figure 4 supports the use of modern tree-based ensembles—particularly gradient boosting—for multi-class prediction of trading signals under the proposed feature set and evaluation design. Importantly, the near-alignment of precision, recall, and F1 for XGBoost and LightGBM indicates that their accuracy gains are not driven by a single dominant class but reflect broadly consistent classification quality, which is essential for operational trading decisions.

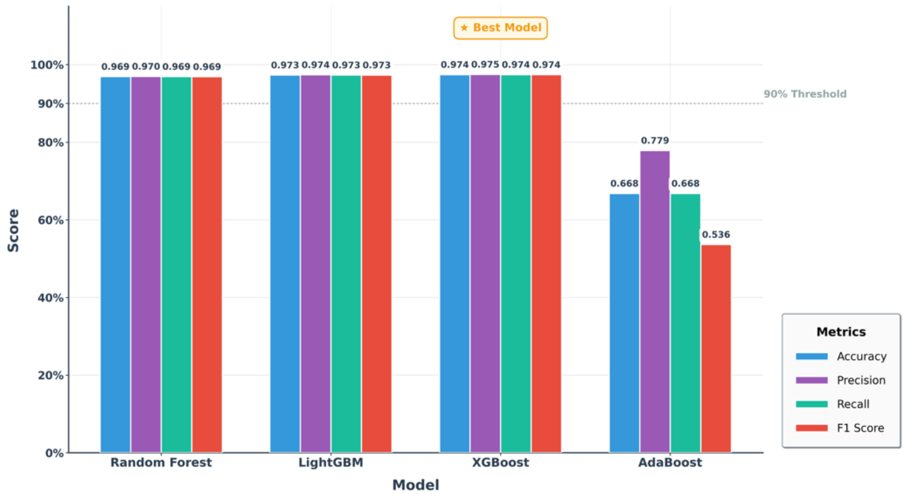


**Figure 4. Comparative Classification Performance of the Evaluated Ensemble Models**

Figure 5 presents the confusion matrices of the four machine learning models—Random Forest, LightGBM, XGBoost, and AdaBoost—used for trading signal classification in this study. The results show that Random Forest, LightGBM, and XGBoost classify the action classes (Buy and Sell) almost perfectly. Buy is essentially error-free (580/580 for LightGBM and XGBoost; 579/580 for Random Forest), and Sell is also very clean (574–577 correct, with only a few cases drifting to Hold). This is important because Buy/Sell are the trades, and consistent recognition of these regimes supports more stable decision-making.

Most remaining errors for these strong models come from Hold, which is naturally the hardest class because it represents "no clear signal." Some Hold days are predicted as Buy or Sell, but the counts stay limited (e.g., for XGBoost: 17 Hold→Buy and 23 Hold→Sell).

AdaBoost, however, has a very different profile. It predicts Buy and Sell perfectly, but it cannot detect Hold (only 2 Hold days are correctly classified; most are misclassified as Buy or Sell). Economically, this is problematic: misclassifying Hold as Buy/Sell means too many trades, and once transaction costs, slippage, and bid–ask spreads are applied, these extra trades can erase paper profits and increase drawdowns.

In short, the confusion matrices suggest that LightGBM and XGBoost are not only accurate, but also more economically plausible, because they do not treat the market as an "always-trade" environment—unlike AdaBoost.

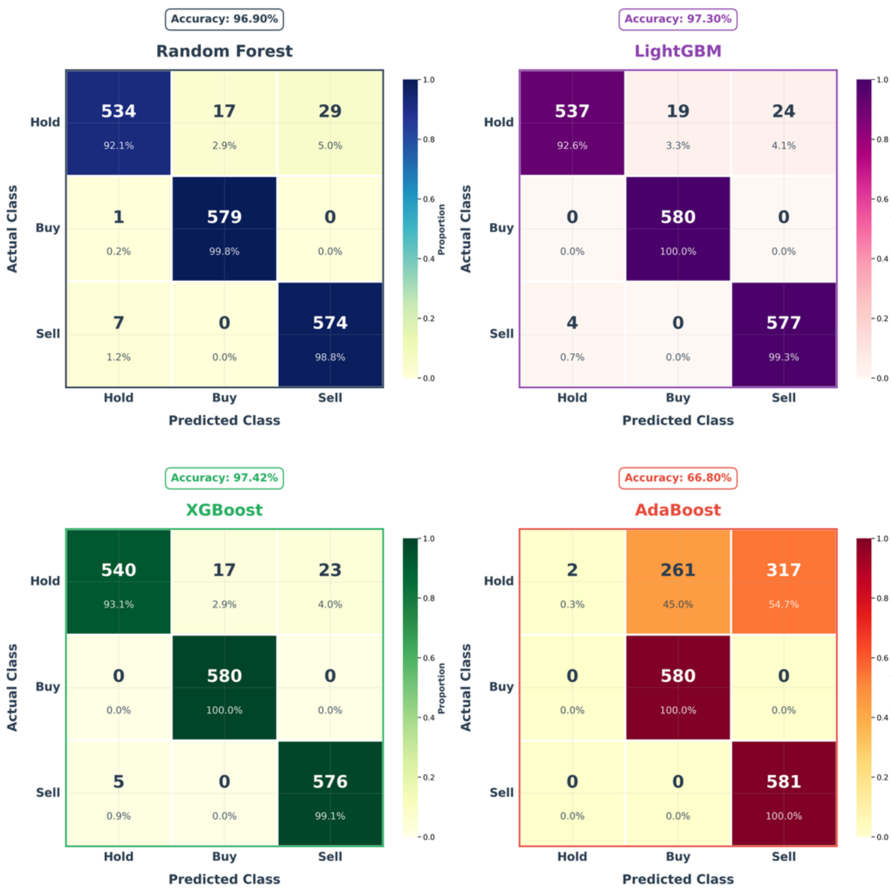


**Figure 5. Confusion Matrices of The Four Classifiers For the Three-Class Trading-Signal Prediction Task**

Figure 6 reports the out-of-sample portfolio value paths produced by trading on each model's predicted signals, starting from an initial capital of $100,000. Three models generate economically meaningful gains over the test window. XGBoost finishes highest at about $149.1K (+49.1%), followed by LightGBM at $146.1K (+46.1%) and Random Forest at $144.9K (+44.9%). Their curves exhibit step-like jumps, as expected in a rule-based long/flat setting: value typically changes more when the model switches positions and captures sustained price moves.

In contrast, AdaBoost ends at roughly $88.7K (−11.3%), indicating that its signals lead to poor trading decisions when realistic frictions are present. Economically, this pattern is consistent with a model that trades too often or enters/exits at the wrong times—exactly the kind of behavior that increases effective costs (commission, slippage, spread) and amplifies drawdowns. Overall, Figure 6 suggests that XGBoost and LightGBM are not only strong on classification metrics but also more viable from a trading perspective, as their predictions translate into higher net wealth under the backtest assumptions.

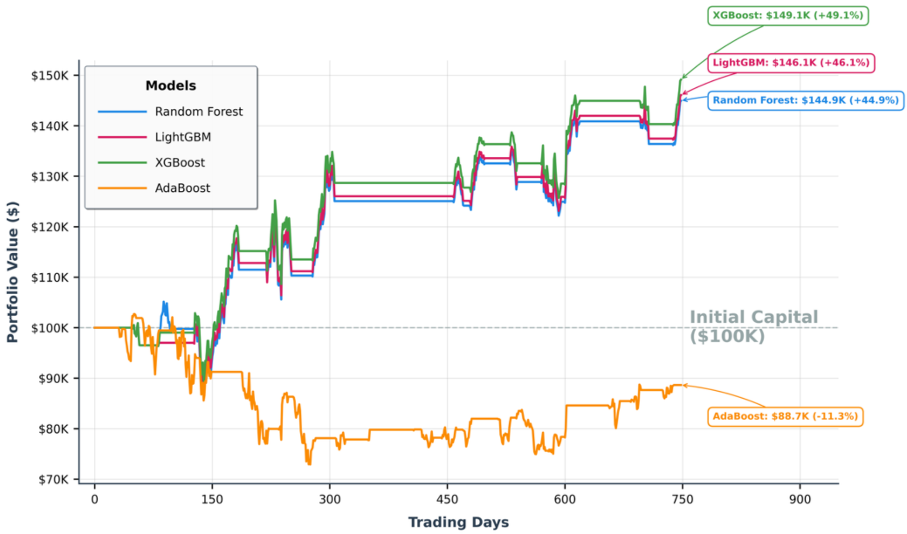


**Figure 6. Portfolio Value Backtest for Ensemble Models**

Figure 7 summarizes the out-of-sample backtest results using the model-predicted signals on the original test period, under the transaction-cost setting (0.34% round-trip; initial capital $100,000). In other words, this figure translates classification outputs into economic performance, which is the key test of whether "high accuracy" also means "useful in trading."

Three models—Random Forest, LightGBM, and XGBoost—produce consistent and economically meaningful gains after costs. Their total returns are +44.9%, +46.1%, and +49.1%, respectively, and the risk-adjusted performance is also positive (Sharpe ≈ 0.86–0.92). Importantly, both LightGBM and XGBoost keep maximum drawdown relatively contained (13.2%), suggesting a more stable equity curve than Random Forest (17.7%).

AdaBoost is the clear outlier. Even though it shows a higher win rate (58.8%), it delivers a negative total return (-11.3%), a negative Sharpe ratio (-0.10), and the largest drawdown (29.0%). The main reason is visible in the trading-activity panels: AdaBoost makes many more trades (68 vs. 23) and therefore pays much higher total transaction costs ($8,761 vs. $4,159). In practical terms, this is a classic case where overtrading + costs dominate the raw "hit rate," so the strategy becomes economically unattractive.

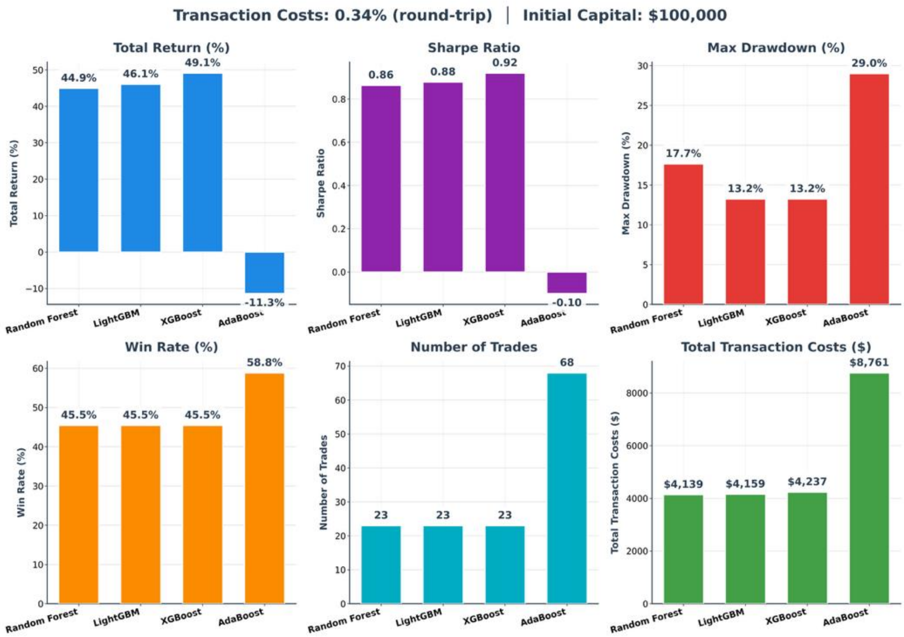


**Figure 7. Backtest Summary Under Realistic Transaction Costs**

After documenting out-of-sample trading performance in Figure 7, Figure 8 helps explain why the models behave as they do by showing the relative importance of the input features in the final classifier. The ranking is dominated by AAPL-derived technical indicators, indicating that the model primarily learns the trading-signal structure from price-based momentum and timing information, rather than from cross-asset levels alone.

The most influential variable is RSI (importance ≈ 0.252), followed by the MACD signal line (≈ 0.135) and MACD itself (≈ 0.098). This is economically intuitive because the target labels are defined directly through MACD vs. signal-line interactions and an RSI filter, so these features carry the strongest immediate information about the signal state. The next two contributors—Volatility (≈ 0.042) and Momentum (≈ 0.036)—suggest that the model also conditions its decisions on risk regime and speed of price changes, which is consistent with trading being harder during turbulent periods.

Cross-asset variables (e.g., VIX, oil, indices, FX) still appear in the list, but their importance is more modest, implying they play a supporting role—mainly refining decisions at the margin—rather than driving the core buy/sell classification.

The model's decisions are anchored in market-timing indicators (RSI and MACD family), while broader macro/market proxies provide incremental context. This supports the interpretation that the strongest predictive content comes from AAPL's own momentum and regime features, which also aligns with the backtest evidence showing that disciplined timing (without excessive turnover) is crucial once transaction costs are included.

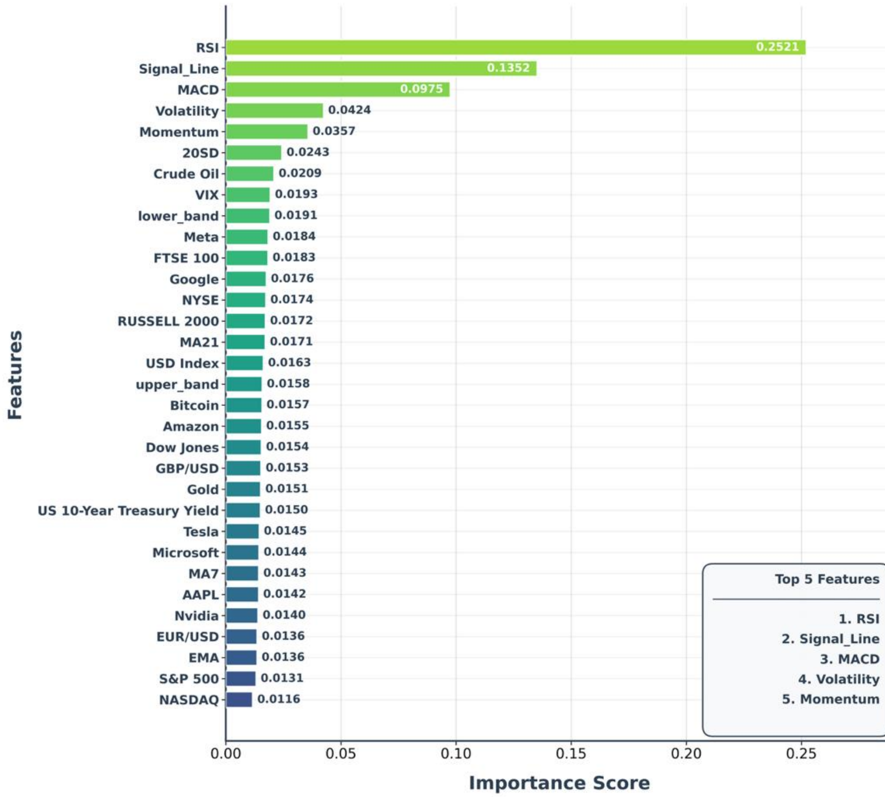


**Figure 8. Feature-Importance Ranking for Trading-Signal Prediction**

## CONCLUSION

This study developed and tested an end-to-end machine learning framework to predict three trading signals (Buy, Hold, Sell) for Apple stock by combining cross-asset market information with AAPL-based technical indicators. The dataset was carefully synchronized to avoid implicit imputation and look-ahead bias, and the target signal was defined using a transparent rule that links MACD and RSI conditions to trading actions. Because the raw labels were highly imbalanced, SMOTE was used during training to ensure that the models learned all classes rather than only the dominant one.

Overall, the results show that tree-based ensemble methods achieve strong out-of-sample performance when classifying the proposed trading signals. Among the competing models, XGBoost and LightGBM delivered the best overall balance across accuracy, precision, recall, and F1, while Random Forest performed similarly but slightly below the top models. AdaBoost lagged behind, producing less stable classification and weaker trading outcomes, suggesting that its boosting structure was poorly matched to this particular signal design and feature space.

Beyond predictive scores, the economic evaluation strengthens the main message. When the model outputs were translated into a simple trading simulation with realistic transaction costs, the top models produced sizable positive portfolio growth, whereas AdaBoost underperformed materially. This gap highlights an important practical point: in trading applications, high classification accuracy is not sufficient on its own—models must also generate signals that lead to consistent decisions under costs and drawdowns.

The feature-importance analysis provides a clear interpretation of what drives predictions. The most influential inputs were RSI and the MACD family (MACD and its signal line), followed by volatility and momentum measures. This is economically intuitive because these indicators summarize trend and timing information directly relevant to rule-based signal states. Cross-asset variables contributed, but mostly as supporting context rather than the primary decision drivers.

This study has several limitations. First, the target labels are rule-based and therefore reflect the chosen MACD–RSI logic; different labeling rules could change both class balance and difficulty. Second, SMOTE improves learning balance but may introduce synthetic patterns that do not fully reflect real market dynamics, so using the original time-ordered test set for backtesting remains essential. Third, the trading simulator uses a simplified long-only position logic and does not explore leverage, short-selling constraints, or more advanced execution models.

Future work can extend the framework in three directions. First, alternative label definitions and threshold sensitivity analysis can test how robust the conclusions are to signal design. Second, richer information sets (e.g., macro surprises, earnings events, sentiment indicators) may improve generalization beyond technical signals. Third, more realistic trading rules—position sizing, stop-loss logic, and cost-aware signal filtering—could improve the link between predictive performance and economic value.

In summary, the evidence supports using optimized gradient-boosting models to learn technical-signal states from a structured feature set. It also demonstrates that combining transparent signal construction with cost-aware backtesting is a practical way to assess whether predictive models can lead to economically meaningful results.

# BAYESYEN OPTİMİZASYONLU TOPLULUK ÖĞRENMESİ İLE ÇOK SINIFLI ALIM-SATIM SİNYALİ SINIFLANDIRMASI

## GENİŞLETİLMİŞ ÖZET

## 1. GİRİŞ

Finansal piyasalarda alım–satım kararlarının zamanlaması; yüksek oynaklık, doğrusal olmayan ilişkiler ve gürültülü veri yapısı nedeniyle zor bir tahmin problemidir. Klasik teknik analiz yaklaşımları, tekil göstergeler üzerinden kural tabanlı sinyaller üretse de, çoklu veri kaynaklarını (çapraz varlık göstergeleri, endeksler, emtialar, kur ve risk göstergeleri) birlikte kullanma ve karmaşık etkileşimleri yakalama konusunda sınırlı kalabilmektedir. Bu çalışma, alım–satım sinyali üretimini üç sınıflı bir sınıflandırma problemi (Al/Tut/Sat) olarak ele alarak, topluluk öğrenme (ensemble) tabanlı modern makine öğrenmesi yöntemlerinin bu görevdeki performansını hem istatistiksel hem de ekonomik açıdan sınamayı amaçlamaktadır.

Çalışmanın temel motivasyonu iki noktada yoğunlaşır: (i) Piyasa verisindeki çok boyutlu ve doğrusal olmayan örüntülerin, ağaç tabanlı topluluk yöntemleri tarafından etkin biçimde öğrenilebilmesi; (ii) "yüksek sınıflandırma başarısı"nın tek başına yeterli olmayıp, üretilen sinyallerin işlem maliyetleri altında gerçekten kârlı ve istikrarlı bir işlem performansına dönüşmesi gerekliliği. Bu nedenle çalışma, model performansını yalnızca Accuracy/F1 gibi ölçütlerle sınırlamamakta; maliyet dâhil backtest ile "ekonomik geçerliliği" de değerlendirmektedir.

## 2. YÖNTEM

Ampirik tasarım, uçtan uca bir tahmin hattı (pipeline) üzerinde kurgulanmıştır: veri toplama–ön işleme, özellik mühendisliği, kural tabanlı hedef değişken (sinyal) üretimi, sınıf dengesizliğinin giderilmesi, model eğitimi/optimizasyonu, test döneminde sınıflandırma ölçütleri ve maliyet dâhil backtest ile ekonomik doğrulama. Veri seti, yfinance aracılığıyla Apple (AAPL) kapanış fiyatları ile ABD hisse endeksleri, seçilmiş teknoloji hisseleri, EUR/USD–GBP/USD kurları, altın–petrol gibi emtialar, VIX, dolar endeksi, Bitcoin ve ABD 10 yıllık tahvil faizi gibi çapraz varlık göstergelerini kapsayacak şekilde oluşturulmuştur. Ham dönem 2014-01-02–2024-12-05 olup, senkronizasyon ve eksik gözlem temizliği sonrasında nihai örneklem 2014-09-17–2024-12-04 aralığında 2519 işlem gününe indirgenmiştir; ayrıca AAPL tabanlı teknik göstergeler (hareketli ortalamalar, Bollinger bantları, MACD, RSI vb.) eklenerek toplam 32 girdili bir özellik uzayı elde edilmiştir.

Hedef değişken, iki yaygın momentum göstergesi kullanılarak şeffaf bir kural seti ile tanımlanmıştır: MACD'nin sinyal çizgisini yukarı kesmesi ve RSI<50 koşulu "Al", MACD'nin sinyal çizgisinin altına inmesi ve RSI>50 koşulu "Sat", diğer tüm durumlar ise "Tut" olarak etiketlenmiştir. Bu kurgu, tahmin problemini üç sınıflı bir sınıflandırma yapısına dönüştürürken aynı zamanda etiketlerin zaman tutarlılığını korur. Ancak bu yapı doğal olarak dengesiz sınıf dağılımı üretmektedir (Tut sınıfı belirgin biçimde baskındır). Bu nedenle eğitim aşamasında SMOTE ile azınlık sınıflar sentetik olarak artırılmış; dengeleme yalnızca eğitim bölümünde uygulanarak test döneminde ileriye dönük yanlılığın önüne geçilmiştir.

Modelleme aşamasında dört ağaç tabanlı topluluk yöntemi karşılaştırılmıştır: AdaBoost, Random Forest, XGBoost ve LightGBM. Hiperparametre seçiminin keyfîliğini azaltmak ve genelleme performansını güçlendirmek için Bayesçi optimizasyon yaklaşımı Optuna ile uygulanmış; modeller kronolojik bir eğitim–test bölmesi altında (yaklaşık %70 eğitim, %30 test) değerlendirilmiştir. Performans iki katmanda raporlanmıştır: (i) Confusion matrix tabanlı çok sınıflı sınıflandırma ölçütleri (Accuracy, Precision, Recall, F1), (ii) Al/Sat sinyallerinin long-only, all-in/all-out işlem mantığıyla maliyet dâhil simülasyona çevrildiği backtest çıktıları (toplam getiri, al-tut getirisi, Sharpe, maksimum düşüş, işlem sayısı, maliyet yükü vb.).

## 3. BULGULAR

Sınıflandırma sonuçları, modern gradyan artırma tabanlı yöntemlerin (özellikle XGBoost ve LightGBM) en yüksek ve en dengeli başarıyı sağladığını göstermektedir. Test döneminde XGBoost; Accuracy, Precision, Recall ve F1 ölçütlerinde en iyi genel performansı üretmiş, LightGBM ve Random Forest çok yakın değerlerle takip etmiştir. AdaBoost ise genel doğruluk ve özellikle dengeli sınıflandırmayı yansıtan F1 bakımından belirgin biçimde geride kalmıştır. Bu ayrışma, dinamik ve dengesiz yapılı finansal etiketlerde, daha güçlü düzenleme ve esnek karar sınırları sunan gradyan artırma mimarilerinin avantajını işaret etmektedir.

Confusion matrix analizi, en güçlü modellerin işlem kararları açısından kritik olan “Al” ve “Sat” sınıflarını neredeyse hatasız yakaladığını; hataların görece daha çok “Tut” sınıfında kümelendiğini ortaya koymaktadır. Bu bulgu pratik açıdan önemlidir: “Tut” sınıfı doğası gereği belirsiz rejimleri temsil ettiği için, bu sınıftaki sınırlı hata toleransı, modelin “her gün işlem yapma” eğilimine sapmaması bakımından gereklidir. AdaBoost’un temel sorunu tam da burada görünür hâle gelmektedir: “Tut” günlerini yeterince ayırt edemediği için gereksiz işlem üretmekte ve işlem maliyetlerine karşı kırılgan bir strateji profili sergilemektedir.

Ekonomik doğrulama (backtest) katmanı, sınıflandırma başarısının “ekonomik fayda”ya dönüşüp dönüşmediğini açık biçimde ayrıştırmıştır. XGBoost, LightGBM ve Random Forest; başlangıç sermayesini test döneminde anlamlı biçimde artırmış; XGBoost en yüksek portföy değerine ulaşmıştır. Buna karşılık AdaBoost, işlem maliyetleri dâhil edildiğinde negatif toplam getiri üretmiş; daha yüksek işlem sayısı nedeniyle toplam maliyet yükü artmış ve maksimum düşüş derinleşmiştir. Bu sonuç, finansal uygulamalarda “yüksek doğruluk” söyleminin tek başına yeterli olmadığını; işlem sıklığı, maliyet ve drawdown gibi unsurların model seçiminin ayrılmaz parçası olduğunu göstermektedir.

## 4. TARTIŞMA

Bulgular, iki temel tartışma noktasını güçlendirmektedir. Birincisi, gradyan artırma tabanlı topluluk yöntemleri (XGBoost/LightGBM), çapraz varlık göstergeleri ve teknik göstergelerden oluşan zengin özellik uzayında; etkileşimleri, eşik davranışlarını ve doğrusal olmayan örüntüleri daha etkin yakalayarak daha istikrarlı sınıflandırma üretmektedir. İkincisi, ekonomik performansın belirleyicilerinden biri "işlem disiplini"dir: Tut rejimini makul düzeyde tanımlayamayan bir model, yüksek işlem devir hızıyla maliyetleri büyütür ve kârlılığı aşındırır. AdaBoost'un kâğıt üzerindeki bazı sınıf doğrularına rağmen ekonomik olarak zayıf kalması, bu mekanizmanın somut bir örneğidir.

Model davranışını açıklamak amacıyla raporlanan önem/katkı analizi, kararların büyük ölçüde AAPL'ye özgü teknik göstergeler tarafından sürüklendiğini göstermektedir. Özellikle RSI ve MACD ailesi (MACD, sinyal çizgisi) en etkili değişkenlerdir; bunu volatilite ve momentum gibi rejim göstergeleri izlemektedir. Bu sonuç ekonometrik olarak da tutarlıdır: Hedef etiketler MACD–RSI mantığıyla üretildiğinden, bu değişkenlerin bilgi içeriği doğrudan yüksektir; çapraz varlık göstergeleri ise daha çok marjinal düzeltme/bağlam sağlama rolünde kalmaktadır. Dolayısıyla çalışma, "sinyal tanımı–özellik uzayı–model çıktısı" arasındaki ilişkiyi şeffaf biçimde görünür kılarak yorumlanabilirliği güçlendirmektedir.

## SONUÇ

Bu çalışma, AAPL için üç sınıflı alım–satım sinyali üretimini; (i) çapraz varlık göstergeleri + teknik göstergelerden oluşan kapsamlı bir özellik seti, (ii) sınıf dengesizliğini gideren SMOTE tabanlı eğitim yaklaşımı, (iii) Optuna ile Bayesçi hiperparametre optimizasyonu ve (iv) işlem maliyetleri altında ekonomik doğrulama (backtest) bileşenleriyle bütünleşik bir çerçevede değerlendirmiştir. Sonuçlar, XGBoost ve LightGBM'nin hem istatistiksel ölçütlerde hem de maliyet dâhil ekonomik performansta en başarılı yöntemler olduğunu; AdaBoost'un ise özellikle "Tut" rejimini ayırt edememesi nedeniyle aşırı işlem üretip maliyetler altında zayıf kaldığını göstermektedir.

Çalışmanın başlıca katkısı, alım–satım sinyali sınıflandırmasında model karşılaştırmasını yalnızca tahmin başarısı ile sınırlamayıp, doğrudan ekonomik uygulanabilirlik testine bağlamasıdır. Gelecek araştırmalar, sinyal etiketleme kuralının (eşiklerin) duyarlılık analizini yaparak etiket tasarımının sonuçlara etkisini ölçebilir; makroekonomik sürprizler ve duygu analizi gibi gerçek zamanlı değişkenleri özellik setine ekleyerek genellemeyi güçlendirebilir; ayrıca pozisyon büyüklüğü, risk bütçeleme ve maliyet-duyarlı sinyal filtreleme gibi daha gerçekçi işlem kurallarıyla strateji katmanını geliştirebilir.

## REFERENCES

Akiba, T., Sano, S., Yanase, T., Ohta, T., and Koyama, M. (2019). Optuna: A next-generation hyperparameter optimization framework. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '19)*, 2623–2631. doi:10.1145/3292500.3330701

Appel, G. (1979). *The Moving Average Convergence-Divergence Trading Method*. Signalert Corporation.

Aroussi, R. (2024). *yfinance* (Version 0.1.70) [Software]. Zenodo. https://doi.org/10.5281/zenodo.13340981

Bollinger, J. (2002). *Bollinger on Bollinger Bands*. McGraw-Hill.

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5–32. https://doi.org/10.1023/A:1010933404324

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). *SMOTE: Synthetic Minority Over-sampling Technique*. *Journal of Artificial Intelligence Research, 16*, 321–357. https://doi.org/10.1613/jair.953

Chen, T., and Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794. San Francisco, CA, United States. https://doi.org/10.1145/2939672.2939785

Cheng, L., Huang, Y., Hsieh, M., and Wu, M. (2021). A novel trading strategy framework based on reinforcement deep learning for financial market predictions. *Mathematics*, 9(23), 3094. https://doi.org/10.3390/math9233094

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5), 1189–1232. https://doi.org/10.1214/aos/1013203451

Freund, Y., and Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1), 119–139. https://doi.org/10.1006/jcss.1997.1504

Gupta, V., and Kumar, E. (2023). H3O-LGBM: Hybrid Harris Hawk Optimization-based Light Gradient Boosting Machine model for real-time trading. *Artificial Intelligence Review*, 56(8), 8697–8720. https://doi.org/10.1007/s10462-022-10323-0

Ji, G., Yu, J., Hu, K., Xie, J., and Ji, X. (2022). An adaptive feature selection schema using improved technical indicators for predicting stock price movements.

*Expert Systems with Applications, 200*, 116941. https://doi.org/10.1016/j.eswa.2022.116941

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., and Liu, T.-Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 3149–3157. Long Beach, CA, United States.

Lemaître, G., Nogueira, F., and Aridas, C. K. (2017). imbalanced-learn: A Python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of Machine Learning Research*, 18(17), 1–5. Retrieved December 8, 2024, from https://imbalanced-learn.org/stable/

Li, Z., and Tam, V. (2018). A machine learning view on momentum and reversal trading. *Algorithms*, 11(11), 170. https://doi.org/10.3390/a11110170

Lin, H., Chen, C., Huang, G., and Jafari, A. (2021). Stock price prediction using generative adversarial networks. *Journal of Computer Science*, 17(3), 188–196. https://doi.org/10.3844/jcssp.2021.188.196

Saifan, R., Sharif, K., Abu-Ghazaleh, M., and Abdel-Majeed, M. (2020). Investigating algorithmic stock market trading using ensemble machine learning methods. *Informatica*, 44(3). https://doi.org/10.31449/inf.v44i3.2904

Saud, A., and Shakya, S. (2022). Directional movement index-based machine learning strategy for predicting stock trading signals. *International Journal of Electrical and Computer Engineering (IJECE)*, 12(4), 4185–4194. https://doi.org/10.11591/ijece.v12i4.pp4185-4194

Sebastião, H., and Godinho, P. (2021). Forecasting and trading cryptocurrencies with machine learning under changing market conditions. *Financial Innovation*, 7(1). https://doi.org/10.1186/s40854-020-00217-x

Wang, Q., Kang, K., Zhihan, Z., and Cao, D. (2021). Application of LSTM and Conv1D LSTM network in stock forecasting model. *Artificial Intelligence Advances*, 3(1), 36–43. https://doi.org/10.30564/aia.v3i1.2790

Wang, Y., and Yan, K. (2023). Application of traditional machine learning models for quantitative trading of Bitcoin. *Artificial Intelligence Evolution*, 4(1), 34–48. https://doi.org/10.37256/aie.4120232226

Wilder, J. W., Jr. (1978). *New Concepts in Technical Trading Systems*. Trend Research.

| **KATKI ORANI /** ***CONTRIBUTION RATE*** | **AÇIKLAMA /** ***EXPLANATION*** | **KATKIDA BULUNANLAR /** ***CONTRIBUTORS*** |
|---|---|---|
| Fikir veya Kavram / *Idea or Notion* | Araştırma hipotezini veya fikrini oluşturmak / *Form the research hypothesis or idea* | Cemal ÖZTÜRK |
| Tasarım / *Design* | Yöntemi, ölçeği ve deseni tasarlamak / *Designing method, scale and pattern* | Cemal ÖZTÜRK |
| Veri Toplama ve İşleme / *Data Collecting and Processing* | Verileri toplamak, düzenlenmek ve raporlamak / *Collecting, organizing and reporting data* | Cemal ÖZTÜRK |
| Tartışma ve Yorum / *Discussion and Interpretation* | Bulguların değerlendirilmesinde ve sonuçlandırılmasında sorumluluk almak / *Taking responsibility in evaluating and finalizing the findings* | Cemal ÖZTÜRK |
| Literatür Taraması / *Literature Review* | Çalışma için gerekli literatürü taramak / *Review the literature required for the study* | Cemal ÖZTÜRK |