ML_SmartMomentum: A Machine Learning-Enhanced Momentum Trading Strategy

Abstract

This study presents ML_SmartMomentum, a systematic approach that integrates technical momentum indicators with a Random Forest machine learning classifier to forecast next-day stock price direction. Data were obtained from Alpha Vantage and subjected to rigorous preprocessing. Key features—Relative Strength Index (RSI), 20-day Exponential Moving Average (EMA20), and Moving Average Convergence Divergence (MACD)—were engineered. The classifier, trained on over 25,000 samples from five equities, yielded an accuracy of 50.3%. A backtesting framework was developed to evaluate strategy performance against an equal-weighted benchmark. Results are visualized in cumulative return charts and analyzed under log-scale representation. Implications for future enhancements are discussed.

1. Introduction

Momentum-based trading strategies rely on the persistence of asset price trends. This work augments traditional momentum indicators with machine learning to improve predictive capabilities. By leveraging Random Forest classification, the model aims to generate robust buy/hold signals based on technical factors.

2. Data Acquisition

Daily historical price data were retrieved via the Alpha Vantage API, covering the period from 2000 to 2025 for five liquid equities: AAPL, MSFT, GOOG, NVDA, and TSLA. The raw dataset comprised over 25,000 observations. Data were cleaned to ensure completeness and consistency, including handling of missing values and chronological sorting.

3. Feature Engineering

Three technical indicators were computed: RSI to measure momentum strength, EMA20 to capture short-term trends, and MACD to identify momentum crossovers. All indicators were implemented using pandas and numpy, preserving date indices for alignment.

4. Machine Learning Methodology

A Random Forest classifier was selected due to its robustness against overfitting and capability to model non-linear relationships. The input feature set consisted of RSI, EMA20, and MACD values for each trading day, and the target variable indicated whether the closing price increased the following day. Model training and evaluation were conducted using scikit-learn, with 70/30 train-test split and accuracy as the primary metric.

5. Backtesting and Evaluation

The trained classifier generated daily trading signals, which were applied to compute strategy returns by multiplying the signal with the next-day price change. An equal-weighted benchmark strategy was implemented for comparative analysis. Cumulative returns were plotted on both linear and logarithmic scales to assess growth dynamics.

6. Results

The model achieved an out-of-sample accuracy of 50.3%. Backtest results demonstrated modest outperformance over the benchmark when examined on a log scale. Notably, cumulative returns exhibited exponential growth, highlighting the impact of compounding.

7. Conclusion

ML_SmartMomentum provides a comprehensive framework for integrating machine learning into momentum trading strategies. Despite moderate predictive accuracy, the end-to-end pipeline—from data acquisition through backtesting—offers a solid foundation for future enhancements, including incorporation of additional features, advanced models, and realistic transaction cost modeling.

References

1. Pedregosa et al., Scikit-learn: Machine Learning in Python, Journal of Machine Learning Research, 2011.

2. McKinney, Wes. Python for Data Analysis: Data Wrangling with pandas, O'Reilly, 2017.

3. Murphy, John J. Technical Analysis of the Financial Markets, New York Institute of Finance, 1999.