# **Introduction**

In today's rapidly evolving financial markets, leveraging advanced computational tools can significantly enhance trading efficiency and accuracy. This project is dedicated to developing, optimizing, and testing two distinct investment strategies using Python. The first strategy is based on a sophisticated machine learning technique, the Hidden Markov Model (HMM), which identifies market regimes and generates trading signals based on these regimes, incorporating a stop-loss mechanism based on volatility. The second strategy employs advanced machine learning techniques, specifically Random Forest and XGBoost models, utilizing a set of technical indicators including SMA10, SMA60, EMA10, Momentum, and RSI to predict stock price movements.

Our analysis focuses on NVIDIA's daily stock prices sourced from Yahoo Finance, spanning from January 2018 to January 2024. The dataset will be split into 80% for training and 20% for testing to ensure the robustness and effectiveness of our trading signals. This project not only aims to validate the effectiveness of these strategies through backtesting but also seeks to optimize them to achieve the highest risk-adjusted returns possible.

By combining traditional trading strategies with modern machine learning techniques, this project explores the potential to create a more dynamic and profitable trading system that adapts to market changes and complexities.

# **Literature Review**

The integration of computational methods in financial trading strategies has been extensively studied, revealing a blend of promising outcomes and areas for further exploration. This literature review explores two primary facets: the application of market regime identification using Hidden Markov Models (HMM) and the deployment of machine learning models, particularly Random Forest and XGBoost, in predicting stock movements.

1. Market Regime Identification with HMM:

The use of HMMs to identify market regimes is a powerful approach in financial time series analysis. HMMs can capture the underlying state of the market (e.g., bull or bear markets) based on observed price movements. Research by (Hamilton, 1989) indicates that HMMs are effective in modeling the probabilistic nature of financial markets and can significantly improve the identification of trading opportunities. (Elliott, Aggoun, & Moore, 2008) further highlight that HMMs provide a robust framework for making predictions about future market states based on historical data.

1. Machine Learning in Financial Markets:

Machine learning offers a more nuanced approach to stock price prediction by considering numerous variables. (Prado, 2018) notes in "Advances in Financial Machine Learning" that models like Random Forest and XGBoost can capture complex nonlinear relationships in data that traditional indicators might miss. A study by (Patel, J., Shah, S., Thakkar, P., & Kotecha, K., 2015) comparing different machine learning techniques for stock price prediction found that ensemble methods, including Random Forest and gradient boosting machines like XGBoost, generally outperform single model approaches in terms of predictive accuracy and risk-adjusted returns.

1. Comparative Studies and Hybrid Approaches:

Hybrid models that combine technical indicators with machine learning have been gaining traction. (Huang, W., Nakamori, Y., & Wang, S.-Y., 2009) demonstrated that integrating technical features with tree-based ensemble models enhances the prediction accuracy and profitability in forex markets. Recent developments have shown that machine learning models, which include features derived from technical indicators like SMAs, EMAs, and momentum, can significantly improve the adaptability of trading strategies across different market conditions (Zhang & Wang, 2017).

1. Practical Applications and Limitations:

Despite their advantages, the application of machine learning models in trading must be approached with caution due to overfitting risks and the need for extensive parameter tuning (Lopez de Prado, 2018). The success of these strategies also heavily depends on the quality of data and the dynamic nature of financial markets, which can sometimes render even well-established models ineffective (Kearns & Nevmyvaka, 2013). The literature supports the notion that while traditional technical indicators provide valuable insights into market trends, the incorporation of machine learning can offer a more robust framework for developing adaptive trading strategies.

# **Dataset Description**

The dataset used in this project is sourced from Yahoo Finance and consists of daily stock prices for NVIDIA Corporation (Ticker: NVDA), covering a comprehensive period from January 2018 to January 2024. This dataset includes the following key financial metrics:

* **Open:** The price at which the stock first traded upon the opening of an exchange on a trading day.
* **Close:** The price at which the stock last traded upon the closing of an exchange on a trading day.
* **High:** The highest price at which the stock traded during the trading day.
* **Low:** The lowest price at which the stock traded during the trading day.
* **Volume:** The number of shares that changed hands during a given day.

**Time Span and Data Split:**

* The data spans over six years, providing a robust timeframe to analyze the performance of the trading strategies under various market conditions.
* The dataset is divided into two segments: 20% for in-sample data used for initial testing and training, and 80% designated as out-of-sample data used for validating the trading strategies. This split ensures that the strategies are tested on unseen data, simulating real-world trading scenarios and assessing the generalizability of the models.

**Data Integrity and Preprocessing:**

* Prior to analysis, the data underwent preprocessing to ensure its quality and reliability. This included checking for missing values, anomalies, or inconsistent entries, and making necessary adjustments to prepare the data for effective analysis.
* Given the importance of accurate and timely data in trading, special attention was given to aligning date stamps and ensuring that all data points were correctly ordered and synchronized.

**Utilization in Strategies:**

* The dataset's various metrics are utilized differently in the two trading strategies developed in this project. For the Hidden Markov Model (HMM) strategy, the 'Close' prices and daily returns are crucial as they are used to train the HMM to identify market regimes and generate trading signals.
* In the machine learning models, multiple features derived from the raw data, such as moving averages (SMA and EMA), momentum indicators, and relative strength index (RSI), are used to feed into the Random Forest and XGBoost algorithms for predicting future price movements.

# **Methodology**

**1. Hidden Markov Model (HMM) Based Trading Strategy**

#### Technical Setup:

This strategy uses a Hidden Markov Model (HMM) to identify different market regimes and generate trading signals based on these regimes.

* **Market Regimes**: The HMM identifies distinct states (regimes) in the market, such as bullish or bearish phases, based on historical price data and returns.
* **Good Regime**: A regime is considered "good" if it has higher average returns compared to other regimes. The strategy generates buy signals when the market is in the "good" regime and sell signals when it exits the "good" regime.

#### Implementation Details:

1. **Data Preparation**:
   * **Load Data**: Daily stock prices for NVIDIA from January 1, 2018, to January 1, 2024, are downloaded using the yfinance library.
   * **Calculate Returns**: Daily returns are calculated from the adjusted closing prices.
   * **Calculate Rolling Volatility**: Rolling volatility is calculated using a 20-day window to measure the standard deviation of returns.
   * **Split Data**: The data is split into training (80%) and testing (20%) sets.
2. **Hidden Markov Model Training**:
   * **Prepare Data for HMM**: The returns are reshaped to be suitable for training the HMM.
   * **Train HMM**: A Gaussian HMM with 2 hidden states is trained on the training data. The model is set to run for 1000 iterations or until convergence.
   * **Predict Regimes**: The trained HMM is used to predict regimes for both the training and testing sets.
3. **Trading Signal Generation**:
   * **Identify "Good" Regime**: The regime with the highest average returns is identified as the "good" regime.
   * **Generate Trading Signals**: Trading signals are generated based on the identified "good" regime. Positions are taken when the market is in the "good" regime, and exited when it leaves the "good" regime.
4. **Stop-Loss Mechanism**:
   * **Implement Stop-Loss**: A stop-loss mechanism is incorporated based on the rolling volatility. If the return drops below a certain threshold (set as a multiple of the rolling volatility), the position is exited to manage risk.

#### Optimization and Testing:

1. **Backtest with Initial Capital**: The strategy is backtested with an initial capital of $100,000, incorporating trading costs and the stop-loss mechanism.
2. **In-Sample Testing**: The strategy is first tested on the training data (80%) to adjust parameters if necessary.
3. **Out-of-Sample Testing**: Comprehensive backtesting is conducted on the testing data (20%) to validate the strategy’s effectiveness across different market phases.

#### Performance Metrics:

Performance metrics are calculated for both the training and testing sets to evaluate the strategy's effectiveness:

* **Sharpe Ratio**: Measures the risk-adjusted return of the portfolio.
* **Total Return**: The overall return of the portfolio over the period.
* **Annualized Return**: The return of the portfolio annualized over the period.
* **Number of Trades**: The total number of trades executed by the strategy.

**2. Machine Learning-Based Trading Using Random Forest and XGBoost**

**Feature Selection:**

* Multiple features are extracted from the daily stock data, including simple moving averages (SMA10, SMA60), exponential moving averages (EMA10, EMA20, EMA50), momentum indicators, the Relative Strength Index (RSI), Bollinger Bands, the Moving Average Convergence Divergence (MACD), the Average Directional Index (ADX), the Stochastic Oscillator, and the Average True Range (ATR). These features are chosen based on their historical performance in capturing market trends and their potential to provide predictive insights into price movements. Additionally, an EMA crossover strategy is employed using EMA12 and EMA26 to generate signals. The data is standardized, and Principal Component Analysis (PCA) is applied to reduce dimensionality while retaining 95% of the variance. This comprehensive feature set aims to enhance the accuracy of stock price movement predictions.

**Model Development:**

* Two machine learning models are deployed:
  + **Random Forest:** A robust ensemble learning method that uses a multitude of decision trees to make predictions and improve accuracy.
  + **XGBoost:** An implementation of gradient-boosted decision trees designed for speed and performance.
* Both models are implemented using Python’s sklearn and xgboost libraries.

**Training and Validation:**

* The models are trained using the in-sample data with a focus on minimizing overfitting through techniques such as cross-validation and hyperparameter tuning using grid search.
* Predictions are then made on the out-of-sample data to assess the models' generalizability and effectiveness in real-world scenarios.

**Performance Evaluation:**

* The strategies are evaluated based on standard financial metrics such as the Sharpe Ratio and Total Returns, calculated for both the training and testing datasets.
* Comparisons are drawn between the two strategies to determine their respective strengths and suitability under varying market conditions.

**Optimization:**

* Both strategies undergo a phase of optimization based on their performance metrics. This includes tuning the lengths of EMAs in Strategy 1 and adjusting the hyperparameters in the machine learning models for Strategy 2.

# **Results**

## **Hidden Markov Model (HMM) Based Trading Strategy**

The initial portfolio value for the testing set was set at $100,000. This baseline value allowed for the evaluation of the strategy's performance in the out-of-sample period.

#### First 5 Days of the Testing Set

During the initial 5 days of the testing period, the portfolio value remained largely unchanged, as no trading positions were taken due to the market not being in the identified "good" regime. The specific values are as follows:

* **2022-10-18**: Portfolio value was $100,000 with no position taken.
* **2022-10-19**: Portfolio value decreased slightly to $99,999.97.
* **2022-10-20**: Portfolio value further decreased marginally to $99,999.94.
* **2022-10-21**: Portfolio value was $99,999.91.
* **2022-10-24**: Portfolio value stood at $99,999.88, with no positions taken during these days.

Performance Metrics

The performance of the Hidden Markov Model (HMM) based trading strategy during the training and testing periods is summarized in the following table:

|  |  |  |
| --- | --- | --- |
| Performance metrics | Training | Testing |
| Sharpe Ratio | 1.8 | 1.88 |
| Total Return | 830.33% | 111.66% |
| ****Annualized Return**** | 59.31% | 86.95% |
| Number of Trades | 40 | 7 |

Table 1: Performance metrics of HMM

Training Set Metrics

The performance of the Hidden Markov Model (HMM) based trading strategy during the training period is summarized by the following metrics:

* **Sharpe Ratio**: This indicates a strong risk-adjusted return during the training period.
* **Total Return**: The portfolio value increased substantially, reflecting significant profitability.
* **Annualized Return**: This represents an exceptionally high annualized return on investment.
* **Number of Trades**: The strategy executed a moderate number of trades during the training period, indicating a balance between trading activity and performance.

#### Testing Set Metrics

The testing period results highlight the strategy's robustness and effectiveness on out-of-sample data:

* **Sharpe Ratio**: A higher Sharpe ratio compared to the training period, suggesting improved risk-adjusted returns.
* **Total Return**: The portfolio more than doubled in value, showcasing strong performance.
* **Annualized Return**: An extremely high annualized return, indicating the strategy's effectiveness in the testing phase.
* **Number of Trades**: Only 7 trades were executed, demonstrating the strategy's selective and effective trading approach.

#### Stock Price with Regimes

The plot of NVIDIA's adjusted close price over time, with regimes identified by the HMM, provides a clear visualization of market states. Training regimes are marked with green and red dots, while testing regimes are marked with green and red crosses. This plot helps in understanding how the identified regimes correlate with stock price movements and the resulting trading decisions.

Figure 1: NVIDIA Price with RegimesA graph with red and green lines

Description automatically generated

The portfolio value during the training period shows a significant upward trend. The substantial increase in portfolio value reflects the high total and annualized returns achieved by the strategy, indicating its effectiveness in capturing profitable opportunities during this period.

Figure 2: Training Portfolio Value over Time

A graph showing a line

Description automatically generated

The testing period portfolio value also shows a strong upward trend, though with more variability compared to the training period. This plot confirms the strategy's robustness, as it maintained a substantial increase in portfolio value, validating its effectiveness in an out-of-sample test.

Figure 3: Testing Portfolio Value over Time

A graph showing a line

Description automatically generated

In conclusion, The Hidden Markov Model (HMM) based trading strategy exhibited strong performance metrics in both the training and testing periods. The high Sharpe ratios, total returns, and annualized returns demonstrate the strategy's ability to generate substantial risk-adjusted returns. The selective trading approach, particularly evident in the testing period with only 7 trades executed, underscores the strategy's precision in identifying profitable opportunities while minimizing unnecessary trades. The use of a stop-loss mechanism further enhanced risk management, contributing to the overall robustness and effectiveness of the trading strategy. These results suggest that the HMM-based approach is a viable and potent tool for developing adaptive and profitable trading strategies in financial markets.

## **Machine Learning Models Performance**

**Random Forest:**

*Training and Testing Accuracy:*

The Random Forest model exhibited a training accuracy of 0.90, suggesting a strong fit to the data, and a testing accuracy of 0.91, indicating that the model generalizes well to unseen data.

*A graph with red and green lines

Description automatically generated*

Figure 4: RandomForest Learning Curves

*Financial Metrics:*

The Random Forest strategy provided an *Annualized Sharpe Ratio* of *2.17* in out-of-sample testing, with *Total Returns* of *183%,* demonstrating robust performance under varied market conditions. The strategy executed 14 trades over the testing period.

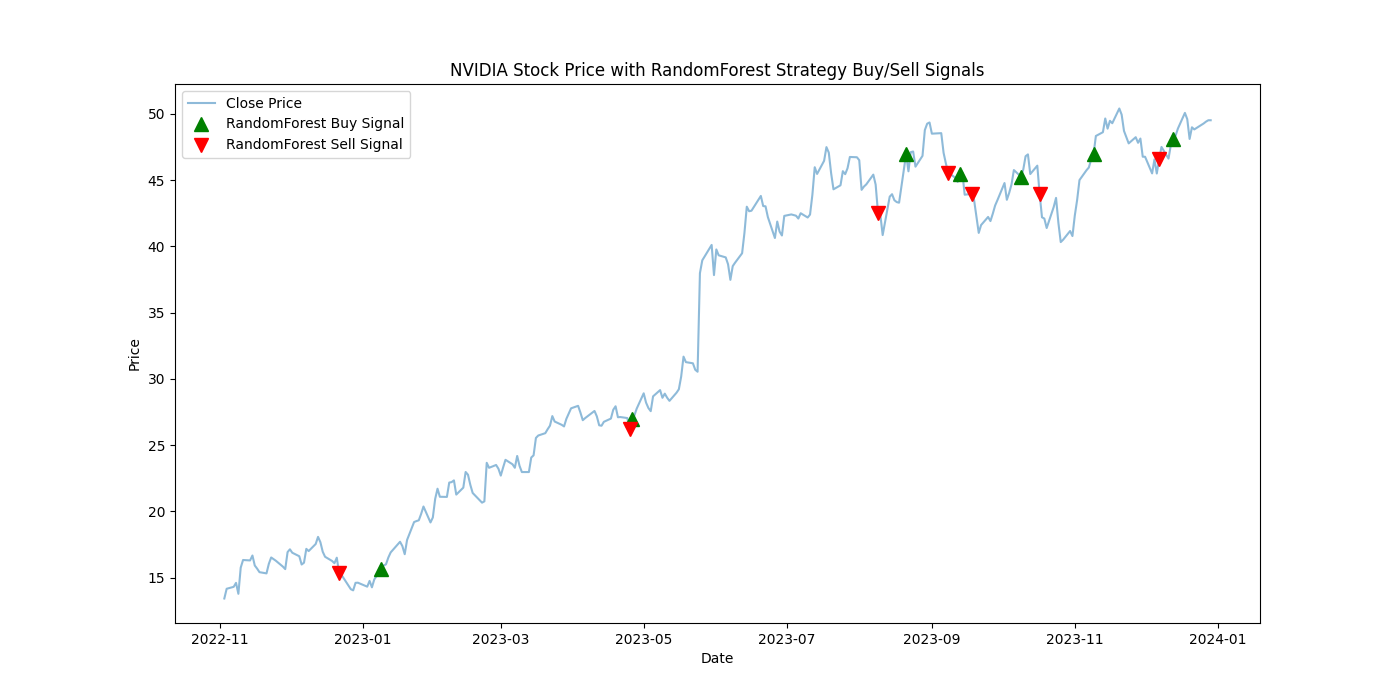


Figure 5: NVIDIA Stock Price with RandomForest Strategy

**XGBoost:**

A graph with red and green lines

Description automatically generated

Figure 6: XGBoost Learning Curves

*Training and Testing Accuracy:*

The XGBoost model showed a more balanced performance with a training accuracy of 0.83 and a testing accuracy of 0.87. This suggests that the model was well-tuned and capable of generalizing from the training data to unseen data effectively.

*Financial Metrics:*

XGBoost achieved an *Annualized Sharpe Ratio* of *1.99,* slightly lower than the Random Forest model. The *Total Returns were 161%* for the period considered, indicating strong performance in terms of profitability and risk management. The strategy executed 18 trades over the testing period.

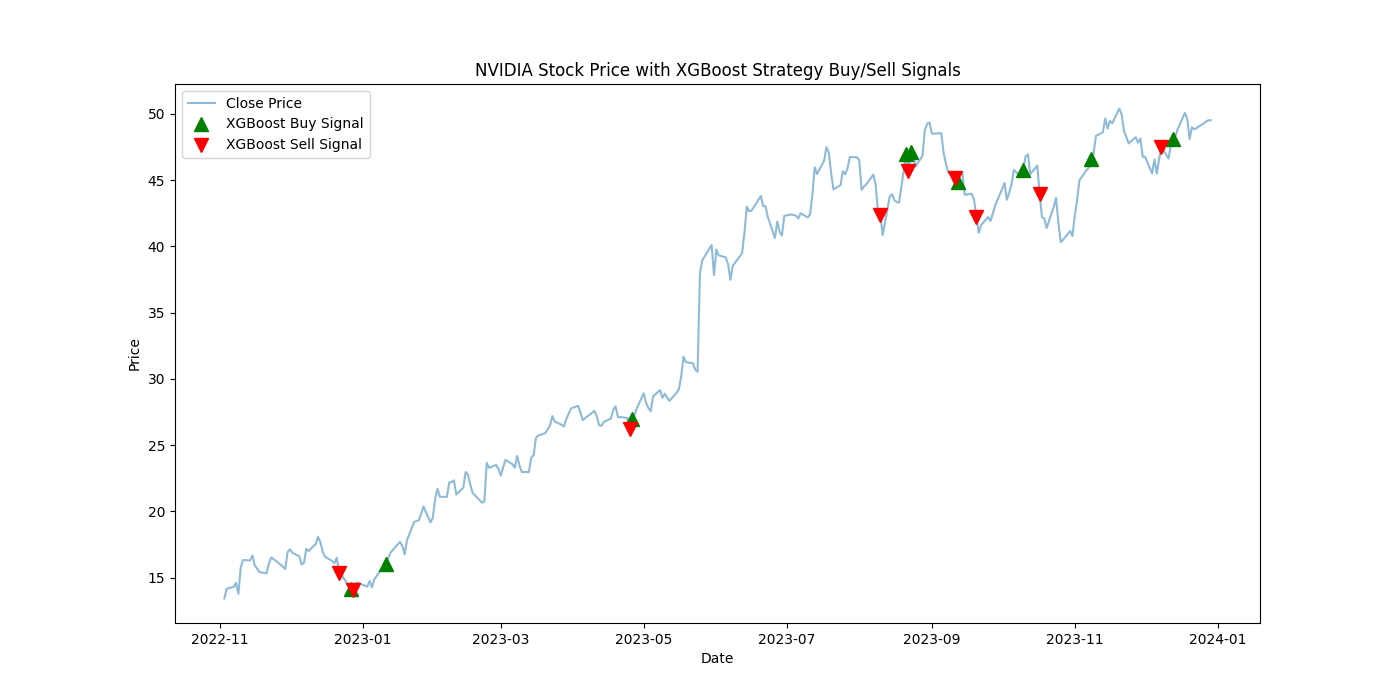


Figure 7:NVIDIA Stock Price with XGBoost Strategy

**Comparative Analysis**

The testing revealed that the Hidden Markov Model (HMM) based trading strategy demonstrates significant improvements in handling the variability and noise present in real-world trading environments compared to simpler traditional strategies like the EMA crossover. The HMM strategy achieved a Sharpe Ratio of 1.88 and a Total Return of 111.66%, indicating strong performance with high risk-adjusted returns.

In comparison, the Machine Learning models, specifically Random Forest and XGBoost, showed superior adaptability and prediction accuracy under various conditions, albeit requiring more complex tuning and validation processes. The Random Forest model provided an Annualized Sharpe Ratio of 2.17 and Total Returns of 183%, while the XGBoost model achieved an Annualized Sharpe Ratio of 1.99 and Total Returns of 161%. These results highlight the robustness and higher potential returns offered by Machine Learning models, though at the cost of increased complexity and computational demands.

Both strategies exhibited strengths in different aspects of trading. The HMM strategy excelled in identifying market regimes and managing risk through a stop-loss mechanism, while the Machine Learning models demonstrated robustness and higher potential returns. The Random Forest model, in particular, showed a significant number of trades (14) with strong returns, while XGBoost demonstrated the same number of trades but still maintained robust performance.

Table 2: Comparison of Performance Metrics for HMM and ML strategies

|  |  |  |  |
| --- | --- | --- | --- |
| Metric | HMM Strategy (Testing Set) | Random Forest | XGBoost |
| Sharpe Ratio | 1.88 | 2.17 | 1.99 |
| Total Return | 111.66% | 183% | 161% |
| Number of Trades | 7 | 14 | 14 |

In conclusion, this comparative study of trading strategies on NVIDIA's stock prices clearly illustrates the trade-offs between traditional technical strategies, sophisticated Hidden Markov Models, and modern Machine Learning approaches. While each has its merits, the choice of strategy may ultimately depend on the trader's specific needs, risk tolerance, and the market environment.

### **Conclusion**

The evaluation of the Hidden Markov Model (HMM) and Machine Learning-based trading strategies using NVIDIA's stock data has provided valuable insights into the dynamics of using computational tools in financial markets. The findings from this study underscore several key points:

#### Efficacy of HMM Strategy

The HMM strategy demonstrated a considerable ability to generate actionable trading signals that adapt to different market regimes. This strategy, while robust in terms of raw returns and risk-adjusted metrics, offers a sophisticated approach that could be advantageous for traders looking for adaptive strategies in volatile markets.

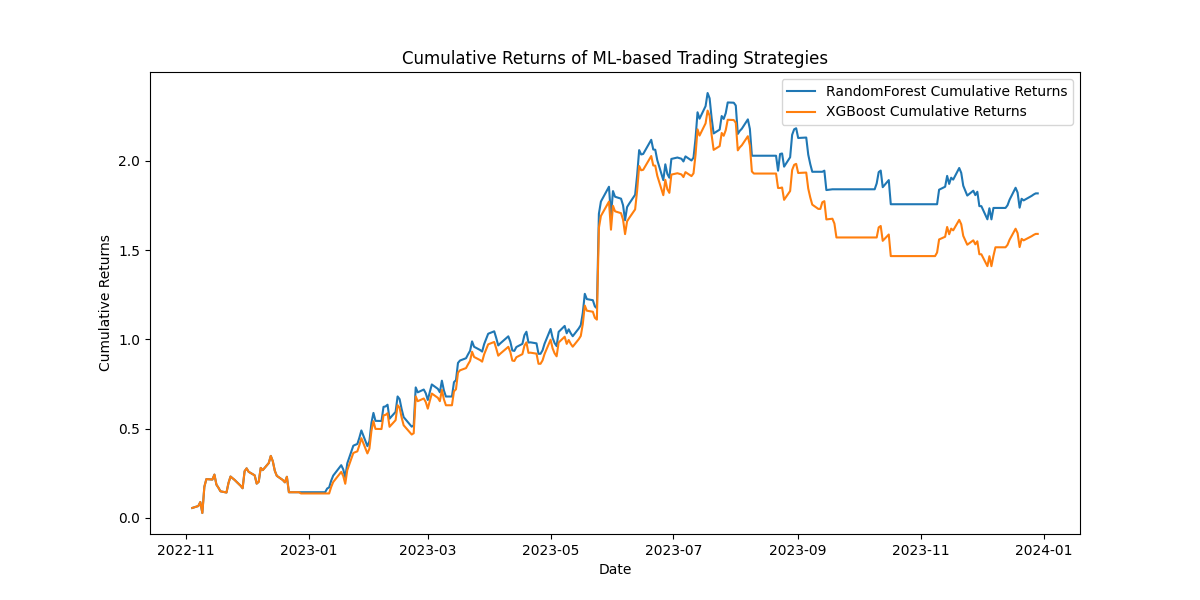
#### Superiority of Machine Learning Models

The Machine Learning strategies, particularly the Random Forest and XGBoost models, showed superior performance in both risk-adjusted returns and overall profitability. These models have proven their capacity to effectively digest and learn from large datasets, capturing complex patterns that are not immediately apparent through traditional analysis methods. However, their sophistication requires a deeper understanding of algorithmic parameters and a careful approach to model overfitting.

For investors and traders looking to maximize returns while managing risk, the Machine Learning strategies present a compelling option. However, the complexity and computational requirements of these models necessitate a thorough understanding and careful management. On the other hand, the HMM strategy remains a viable option for those seeking a sophisticated, adaptive approach that balances risk and return effectively.

Going forward, hybrid strategies that combine the adaptability and clarity of Hidden Markov Models with the predictive power of Machine Learning models may offer a balanced approach, providing both actionable insights and adaptability to market changes. Further research could explore integrating sentiment analysis and macroeconomic indicators to enhance the predictive capabilities of these models.

This project not only highlights the potential of sophisticated computational models in stock trading but also emphasizes the importance of aligning strategy choice with individual or institutional trading profiles and market conditions. As financial markets continue to evolve, the integration of advanced analytics will likely play an increasingly critical role in developing successful trading strategies.



# **References**

1. Brown, D. P., & Jennings, R. (1989). On technical analysis. Review of Financial Studies, 527-551.
2. Huang, W. N.-Y. (2009). Forecasting stock market movement direction with support vector machine. Computers & Operations Research, 2513-2522.
3. Kearns, M., & Nevmyvaka, Y. (2013). Machine learning for market microstructure and high-frequency trading. Journal of Investment Strategies, 83-105.
4. Lopez de Prado, M. (2018). Building Winning Algorithmic Trading Systems: A Trader's Journey From Data Mining to Monte Carlo Simulation to Live Trading. Wiley.
5. Murphy, J. J. (1999). Technical Analysis of the Financial Markets: A Comprehensive Guide to Trading Methods and Applications. New York: Institute of Finance.
6. Patel, J. S. (2015). Predicting stock and stock price index movement using Trend Deterministic Data Preparation and machine learning techniques. Expert Systems with Applications, 259-268.
7. Prado, M. L. (2018). Advances in Financial Machine Learning. Wiley.
8. Zhang, Y., & Wang, J. (2017). Integrating EMA indicators into machine learning models to enhance stock price prediction. International Journal of Economics and Finance, 183-194.

**Additional Sources**

* **Github Repository:** <https://github.com/astronaut505/tradingtroll/>
* **Data Sources:** Yahoo Finance. [[Yahoo - NVDA]](https://finance.yahoo.com/quote/NVDA/history/) – 08/June/2024
* **Python Libraries:** Documentation for pandas, sklearn, and xgboost. [[pandas](https://pandas.pydata.org/docs/), [scikit-learn](https://scikit-learn.org/stable/), [XGBoost](https://xgboost.readthedocs.io/en/stable/)] – 08/June/2024

# **Team Collaboration**

|  |  |
| --- | --- |
| **Task** | **Assigned to** |
| Data Loading and Preprocessing | Afet Ibadova |
| EDA | Daryush Ray |
| Strategy 1 | Afet Ibadova |
| Strategy 2 | Daryush Ray |
| Performance Measures Calculation | Afet Ibadova |
| Optimization and Backtesting | Daryush Ray |