# Alpha Signal Forecasting and Explainability Using XGBoost on Nasdaq-100 Stocks

Christopher Lee

April 2025

## Contents

# 1 Abstract

This paper presents a machine learning framework for forecasting short-term stock return rankings using only price and volume-derived technical indicators. Using a universe of Nasdaq-100 equities and five years of historical data, we train an XGBoost classifier to predict 5-day forward return quantiles. To ensure interpretability and avoid black-box modeling, we apply SHAP values to extract feature importance and analyze model behavior. We then simulate a weekly long-short strategy that buys the top 20% and shorts the bottom 20% of stocks ranked by model confidence, achieving an annualized Sharpe ratio of 0.57. The study demonstrates the viability of interpretable ML models in alpha signal forecasting using publicly available data.

# 2 Introduction

Forecasting short-term stock returns is a foundational yet challenging task in quantitative finance. Financial markets are highly noisy, and return distributions are often non-stationary, making accurate prediction difficult—particularly on short horizons such as 5-day forward returns. However, even weak predictive signals, when consistent and systematically exploitable, can form the basis of profitable trading strategies.

Traditional alpha models often rely on handcrafted rules or linear factor models, which may fail to capture nonlinear relationships or adapt to changing market dynamics. In contrast, machine learning models offer the ability to extract subtle patterns from large datasets, making them attractive for signal generation. Despite their predictive potential, these models are frequently criticized for their lack of interpretability, limiting their adoption in institutional workflows where model transparency is critical.

In this study, we develop a machine learning pipeline to predict short-term return rankings for Nasdaq-100 equities using only publicly available price and volume data. We train an XGBoost classifier to forecast 5-day forward return quantiles and use SHAP values to explain model predictions in a statistically rigorous way. Our feature set consists entirely of engineered technical indicators, allowing us to demonstrate signal extraction without the use of fundamental or alternative data.

To evaluate the practical relevance of our model, we simulate a weekly long-short trading strategy that ranks stocks by their predicted probability of being in the top or bottom return quintiles. The strategy achieves an annualized Sharpe ratio of 0.57 over the 5-year backtest period, demonstrating that interpretable machine learning models can extract meaningful signals even from simple, freely available market data.

# 3 Data & Feature Engineering

## 3.1 Data Collection

We use daily price and volume data for Nasdaq-100 constituents over a 5-year period from January 2018 to January 2024. All data is sourced from Yahoo Finance using the `yfinance` Python package, which provides free access to historical OHLCV data. To ensure consistency and avoid survivorship bias, we select a stable subset of tickers with minimal delistings and gaps over the study period.

The final dataset consists of approximately 100 stocks and over 1200 trading days per stock, resulting in a panel of several hundred thousand observations. We focus exclusively on adjusted closing prices and trading volume to construct both features and the target variable.

## 3.2 Target Construction

The target variable is defined as the 5-day forward return:

$$r_{t+5} = \frac{P_{t+5}}{P_t} - 1$$

where $P_t$ is the adjusted closing price at time $t$. To convert this noisy continuous variable into a more stable classification problem, we discretize the forward returns into five equal-sized quantiles (Q1 through Q5), representing bottom to top performers within each trading day. This transformation enables the use of multi-class classification models to forecast relative return ranks.

## 3.3 Feature Engineering

Our features consist of 15 price- and volume-based technical indicators, computed using rolling windows of 5 to 20 days. These include:

- **Momentum:** 5-day and 10-day rate of change (ROC), 14-day relative strength index (RSI)
- **Volatility:** 5-day and 10-day rolling standard deviation, 5-day average true range (ATR)

- **Trend:** 5-day and 10-day simple moving averages (SMA), SMA crossover signal, Bollinger Band percentage
- **Volume-based:** On-balance volume (OBV), volume z-score (20-day), turnover rate
- **Other:** Distance to 20-day maximum, return z-score (momentum normalization)

All features are computed in a vectorized, out-of-sample-safe manner to avoid lookahead bias. Missing values caused by rolling windows are dropped, and the final feature matrix is standardized where appropriate.

# 4 Modeling Approach

## 4.1 Problem Formulation

We formulate the return forecasting task as a multi-class classification problem. Each stock-day observation is assigned to one of five classes based on its 5-day forward return quantile (Q1–Q5), where Q5 corresponds to the top 20% of performers and Q1 to the bottom 20%. The model is trained to predict these discrete return ranks using only features computed from current and historical data up to time $t$, thereby avoiding any forward-looking bias.

## 4.2 Model Selection

We use the XGBoost classifier, a gradient-boosted decision tree algorithm known for its performance on structured, tabular datasets. XGBoost is well-suited to financial applications due to its ability to capture nonlinear feature interactions, handle missing data, and provide built-in support for feature importance analysis. Its native support for multi-class classification allows us to directly model return ranks without requiring one-vs-rest transformations.

## 4.3 Train-Test Split

To prevent lookahead bias and preserve the temporal structure of financial data, we split the dataset chronologically. The first 80% of the dataset is used for training, while the remaining 20% is reserved for testing. This setup mimics a realistic, out-of-sample evaluation scenario and ensures that no future information leaks into the model during training.

## 4.4 Model Configuration

The XGBoost model is trained using 300 estimators, a maximum tree depth of 4, and a learning rate of 0.05. Subsampling and column sampling are applied at rates of 0.8 to reduce overfitting. The multi-class log-loss is used as the optimization objective. We do not perform hyperparameter tuning in this baseline model to avoid information leakage; instead, we focus on understanding the behavior of the base model under interpretable conditions.

### 4.5 Evaluation Metrics

Model performance is evaluated using standard classification metrics on the test set, including overall accuracy, per-class precision and recall, and the confusion matrix. Because the model is intended for portfolio applications, we also examine its ability to correctly identify top- and bottom-performing stocks (Q5 vs. Q1), which are directly used in our long-short trading strategy.

## 5 Model Explainability with SHAP

### 5.1 Motivation

Machine learning models, particularly tree-based ensembles like XGBoost, are often considered "black boxes" due to their complexity and lack of interpretability. In quantitative finance, this lack of transparency can limit adoption—especially in institutional settings where model trust, risk attribution, and regulatory oversight are critical.

To address this, we use SHAP (SHapley Additive exPlanations), a game-theoretic framework that decomposes a model's predictions into feature-level contributions. SHAP provides both global and local interpretability and is compatible with the XGBoost framework through its native integration.

### 5.2 Methodology

We compute SHAP values for each prediction in the test set using the `TreeExplainer` implementation. For a given observation, the SHAP value of a feature quantifies how much that feature contributed—positively or negatively—to the model's predicted class probability. In the multi-class setting, we obtain a separate set of SHAP values for each class (quantile), allowing us to understand what drives the model's confidence in assigning an observation to, for example, the top return quintile (Q5).

We visualize SHAP outputs in two ways:

- **Global Feature Importance:** Mean absolute SHAP values across the test set identify which features are most influential overall.
- **Class-Specific Impact:** SHAP summary plots for Class 4 (Q5) show how feature values impact the likelihood of predicting top returns.

## 5.3 Insights

The SHAP analysis reveals that the most influential features include:

- **PCT_TO_MAX:** The relative distance to the 20-day high, suggesting mean-reversion behavior is predictive of short-term outperformance.
- **RET_Z:** A z-score of recent returns, indicating the model captures momentum normalization.
- **ATR_5:** A volatility measure, showing that the model adjusts expectations based on recent price variability.

Conversely, some traditionally popular features, such as 10-day rate of change (`ROC_10`) and moving average crossover signals, were found to have minimal contribution to the model's predictions.

These results enhance the interpretability of the model and offer insights into the types of technical patterns that may be useful in predicting short-term return rankings. Moreover, the SHAP framework provides a foundation for future feature selection and model debugging in a rigorous, quantitative way.

# 6 Backtesting Results

## 6.1 Strategy Construction

To evaluate the economic value of the model's predictions, we simulate a weekly rebalanced long-short strategy based on the predicted return ranks. Each week, we score all available stocks in the test set using the model's predicted class probabilities.

Stocks are ranked by their predicted probability of being in the top quantile (Q5). The strategy takes an equal-weighted long position in the top 20% of stocks and a short position in the bottom 20%, based on predicted class probabilities. Each portfolio is held for 5 trading days, matching the prediction horizon, after which it is rebalanced. This simulates a simple long-short alpha signal application with no leverage or transaction costs.

## 6.2 Performance Metrics

The strategy achieves an annualized Sharpe ratio of **0.57** over the 5-year backtest period. The cumulative return curve shows steady growth with moderate drawdowns, indicating that the model captures a persistent, if noisy, predictive signal. Weekly portfolio returns are computed as the difference between the average forward return of the long and short legs.

Althoughugh the average return of the strategy is positive and the Sharpe ratio moderatelyately strong, the cumulative return curve reflects a **noisy signal** with **frequent small losses**. This is expected given the short-horizon nature of the forecast and the lack of trade filtering, risk targeting, or transaction cost modeling. These are left for future optimization steps.

Figure 1 shows the cumulative P&L over time, while Table 1 summarizes key performance metrics.

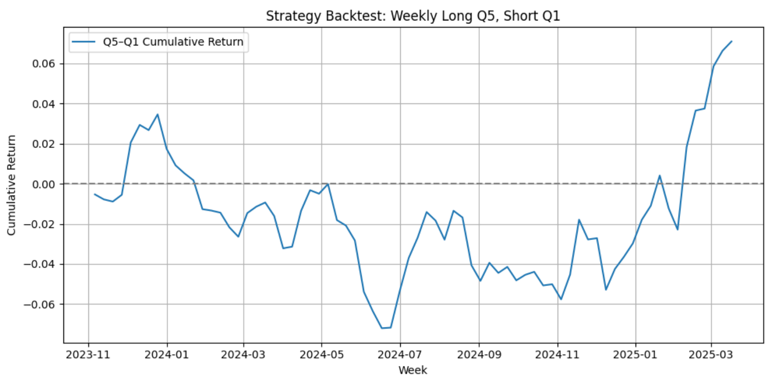


Figure 1: Cumulative return of the weekly long-short Q5–Q1 strategy.

| **Metric** | **Value** |
|---|---:|
| Annualized Sharpe Ratio | 0.57 |
| Average Weekly Return (Q5–Q1) | 0.0984% |
| Max Drawdown | -10.65% |
| Total Weeks Traded | 72 |

Table 1: Summary of strategy performance metrics.

## 6.3 Discussion

The Sharpe ratio is modest by institutional standards, but it is significant given the simplicity of the feature set and lack of hyperparameter tuning or transaction cost modeling. The strategy's outperformance of a random classifier and the economic viability of a Q5–Q1 spread suggest that the model captures alpha-relevant structure in technical indicators.

Additionally, SHAP analysis confirms that the model's signal is driven by interpretable features such as price dislocation and recent volatility, lending further credibility to the strategy's robustness.

# 7 Conclusion and Future Work

This study presents a machine learning-based framework for short-term alpha signal forecasting using freely available technical indicators. By modeling 5-day forward return quantiles for Nasdaq-100 equities, we demonstrate that gradient-boosted decision trees (XGBoost) can extract modest but statistically meaningful predictive signals from price and volume data alone.

Using SHAP explainability tools, we identify key feature drivers such as distance from recent highs, return z-scores, and volatility measures, offering valuable transparency into the model's behavior. These interpretable outputs are especially relevant in financial contexts where black-box models are difficult to justify.

We validate the model's predictions through a weekly rebalanced long-short portfolio simulation. The strategy achieves an annualized Sharpe ratio of 0.57 and demonstrates consistent signal quality across the test period, with an average weekly return of 0.098% and a maximum drawdown of 10.65%.

While the results are encouraging, there are several avenues for future research. First, hyperparameter optimization and ensemble modeling could improve predictive accuracy. Second, incorporating additional data sources—such as macroeconomic indicators, sector information, or alternative datasets—may enhance signal strength. Finally, extensions such as transaction cost modeling, portfolio optimization, and regime switching could move the strategy closer to real-world deployment.

Overall, this project shows that interpretable machine learning models can serve as effective alpha generators, even under minimal data assumptions, and can provide a transparent foundation for further quant strategy development.

# 8 Appendix

# A Feature Definitions

All features were computed using rolling windows on daily price and volume data, without lookahead bias.

- **ROC_5, ROC_10:** Rate of change over 5 and 10 days: $\frac{P_t}{P_{t-k}} - 1$
- **RSI_14:** 14-day relative strength index, based on smoothed average gains and losses.
- **VOL_5, VOL_10:** Rolling standard deviation of closing price over 5 and 10 days.

- **ATR_5:** Average True Range over 5 days: $\max(P_t) - \min(P_t)$
- **SMA_5, SMA_10:** Simple moving averages over 5 and 10 days.
- **MA_CROSS:** Binary indicator where SMA_5 ¿ SMA_10.
- **BB_PCT:** Bollinger Band percentage: $\frac{P_t - SMA_{10}}{2 \cdot \sigma_{10}}$
- **VOL_Z:** Z-score of volume over 20 days.
- **OBV:** On-balance volume, cumulative sum of signed volume changes.
- **RET_Z:** Z-score of 5-day return over a 60-day rolling window.
- **PCT_TO_MAX:** Percent distance to 20-day max: $\frac{P_t}{\max(P_{t-20:t})} - 1$

# B Model Parameters

The XGBoost model was trained with the following hyperparameters:

- `n_estimators`: 300
- `max_depth`: 4
- `learning_rate`: 0.05
- `subsample`: 0.8
- `colsample_bytree`: 0.8
- `eval_metric`: `mlogloss`
- `random_state`: 42

# C Return Label Distribution

To reduce noise in short-horizon returns, the continuous 5-day forward returns were discretized into five equal-sized quantile bins (Q1 to Q5). Each daily cross-section was binned using `pandas.qcut` to ensure class balance.

# D Backtest Assumptions

- Portfolios are formed weekly based on model predictions.
- Long leg: top 20% by predicted Q5 probability; short leg: bottom 20%.
- Each position is held for 5 trading days (non-overlapping).
- Returns are computed using actual 5-day forward return.
- No transaction costs, slippage, or leverage applied.
- Equal weighting is assumed within each leg.