Alpha Ranking Portfolio

François Goybet, Gavriil Kharkhordine, Haocong Li

March 2026

1 Project Summary

1.1 Objective

The goal of this project is to develop a systematic investment strategy in a low to mid latency trading setting, with periodic rebalancing (weekly or monthly). The approach is divided into two main components: (i) a machine learning module designed to generate predictive signals by ranking assets based on their expected relative performance, and (ii) a portfolio optimization module that transforms these signals into portfolio weights while controlling risk and market exposure.

1.2 Universe

The strategy is applied to a cross-sectional universe of approximately 50 to 500 stocks. The portfolio is rebalanced at a weekly or monthly frequency.

1.3 Data

For this project, we plan to collect financial data primarily from WRDS, which provides high-quality structured datasets including historical prices, volumes, and firm fundamentals. As an alternative or supplement, free sources such as Yahoo Finance can be used for basic price and volume data, though with more limited coverage and reliability.

In future iterations, we may explore using Databento to access a wider range of market data, potentially integrated with trading platforms like NautilusTrader for more advanced backtesting or low-latency execution (see discussion below).

The features considered for the model include:

Price data

- Daily or monthly returns
- Trading volume
- Historical volatility
- Momentum measures (e.g., past 12-month return)

Fundamental data

- Price-to-Earnings (P/E)
- Return on Equity (ROE)
- Debt-to-Equity
- Revenue Growth

Market data

- Market index returns
- Risk-free rate
- Market volatility indices

Macroeconomic and geopolitical data

- News sentiment analysis
- Interest rates and inflation indicators
- GDP growth
- Geopolitical risk indices
- Economic policy uncertainty indices

1.4 Module 1: ML Model (Alpha Generation)

The objective of the model is to predict the relative performance of stocks rather than their absolute returns.

The input consists of panel data of the form:

$$(asset_i, t, X_{i,t})$$

where $X_{i,t}$ represents the features available at time t.

A tree-based ensemble model (e.g., gradient boosting) is used to produce predictive scores for each stock:

$$score_{i,t}$$

which are interpreted as alpha signals. Stocks are ranked at each date based on these scores.

Two main types of ranking loss functions can be applied:

- **Pairwise ranking loss:** The model compares pairs of assets and penalizes incorrect ordering. If asset i has a higher realized return than asset j, the loss encourages the predicted score to satisfy $score_i > score_j$. A common implementation uses a logistic function:

  $$L = \log(1 + \exp(-(score_i - score_j)))$$

  This approach is simple, robust, and widely used in tree-based ranking models (e.g., XGBoost with `rank:pairwise`).

- **Listwise ranking loss (NDCG-based):** This loss evaluates the predicted ranking over the entire set of assets at once, focusing on the top of the list. The model is trained to maximize a metric such as Normalized Discounted Cumulative Gain (NDCG), which gives higher weight to correctly ordering the most important assets (e.g., top performers). This approach is more directly aligned with the portfolio construction objective but is more complex and computationally intensive.

By using ranking losses instead of regression losses, the model focuses on predicting *relative* performance rather than exact returns, which is better suited for selecting top/bottom assets for a long/short strategy.

1.5 Module 2: Portfolio Construction

The ranking produced by the model is transformed into a portfolio.

1.5.1 Long-only Strategy

The portfolio is constructed by selecting the top x% (5% at first) of stocks and taking long positions.

1.5.2 Long/Short Strategy

A long/short portfolio is constructed as follows:

- Long: top x% of stocks
- Short: bottom x% of stocks

This approach aims to reduce exposure to market beta.

1.6 Weighting Schemes

Different allocation methods are considered:

- **Equal weights:** Each selected asset is assigned the same weight:

$$w_i = \frac{1}{K}$$

  where K is the number of selected assets. This approach is simple, robust, and avoids estimation errors, but does not account for differences in risk or expected return.

- **Proportional to market capitalization:** Weights are proportional to the market capitalization of each asset:

$$w_i \propto MCAP_i$$

  This gives more importance to larger companies, improving liquidity and reducing turnover, but may dilute the impact of the alpha signal.

- **Inverse volatility (risk-based):** Weights are inversely proportional to the asset's volatility:

$$w_i \propto \frac{1}{\sigma_i}$$

  This allocates more capital to less volatile assets, leading to better risk diversification and more stable portfolios.

- **Kelly-inspired allocation:** We use a probabilistic version of the Kelly criterion at the asset level. For each stock, the optimal fraction is given by:

$$f_i^* = \frac{p_i \cdot b_i - q_i}{b_i}$$

  where p_i is the estimated probability of a positive outcome, $q_i = 1 - p_i$, and b_i represents the payoff ratio.

  In practice, we apply a fractional Kelly approach by scaling:

$$w_i = 0.5 \cdot f_i^*$$

  Negative values ($f_i^* < 0$) are interpreted as short signals. This approach allows the allocation to directly reflect both the strength and confidence of the model's predictions, while reducing sensitivity to estimation errors through scaling.

In practice, a fractional Kelly scaling is used to improve robustness.

1.7 Risk Control

The strategy includes mechanisms to control risk and reduce market exposure, including position sizing and diversification.

1.8 Backtesting

The model is trained on historical data from 2008 to 2018. A backtest is then performed on the out-of-sample period from 2018 to 2025 to evaluate the strategy's performance.

At each rebalancing date:

1. Predict scores for the next period
2. Rank assets according to the predicted scores
3. Construct the portfolio using the chosen weighting scheme
4. Compute realized portfolio returns

The backtest includes realistic considerations such as transaction costs, rebalancing constraints, and turnover limits to produce robust performance metrics.

2 Future Addons

2.1 Order Book Modeling with Nautilus Trader

A natural extension would be to incorporate limit order book data via Nautilus Trader to more realistically model execution costs, accounting for bid-ask spread, market impact, slippage, liquidity constraints, and partial fills, all of which can significantly erode the theoretical alpha identified by the ranking model.

2.2 Event-Driven Time Series Forecasting

The DAFF-Net paper (`https://doi.org/10.1038/s41598-025-22926-y`) proposes a deep learning framework that combines a cross-asset sector correlation matrix with multimodal time series modeling, which could enrich our alpha signals with both inter-stock structural relationships and event-driven market narratives.