Algorithmic Trading Case Packet

1 Overview

Welcome to the Algorithmic Trading Case! Participants will develop an algorithm to allocate capital for a portfolio of 26 stocks, which vary in exchange/country, market cap, and industry. The goal is to create an algorithm that optimally allocates funds over a **4-month horizon** to maximize performance on various financial metrics.

2 Objective

Your objective is to develop an algorithm that constructs and rebalances a portfolio based on daily market data. The performance of your algorithm will be evaluated based on the following metrics:

- **Sharpe Ratio**
- **Sortino Ratio**
- **Max Drawdown**
- **Volatility**
- **Annual Return**
- **Calmar Ratio**
- **Win Rate**

Each algorithm's performance on each of these metrics will be compared individually against others, resulting in each algorithm having a separate rank for each metric. A **weighted sum of the ranks** will determine the final score, where the **lowest weighted sum of ranks wins**.

3 Provided Data

Participants will receive:

- **One year** of historical daily price (in USD) and volume data for each stock
  - Open, Close, High, Low prices
  - Trading Volume
- Example implementation of a valid submission

Both files are attached in the same email as this case packet.

4 Market Assumptions

- No **exchange fees** or **bid-ask spreads**.
- No **liquidity concerns**.
- All weights must be **non-negative**. The backtester will automatically normalize the weights to sum to 1.
- Returns for each stock are based on the % change in closing price between two time steps.

5 Submission Guidelines

- Participants must submit a Python file (.py) on **Gradescope** before the deadline.
- Participants can submit individually or as a part of a team of up to 3 individuals.
- The submission must contain a **class** with an `allocate` function.
  - The backtester will create **1 instance** of your class, after which it will call the `allocate` function of your instance for every time step.
  - Each time it calls `allocate`, the backtester will pass in the trading data for the corresponding time step.
  - The `allocate` function must return a **numpy array** of portfolio weights.
  - All weights must be **non-negative**. The backtester will automatically normalize the weights to 1.
- Participants **can choose to** store useful information (e.g., moving average returns) as instance variables within the class.

6 Rules & Restrictions

- **Programming Language:** Python 3.13
- **Allowed Libraries:**
  - `numpy>=2.2.2`
  - `pandas>=2.2.3`
  - `scipy>=1.15.1`
- Participants can use **any packages or tools** to analyze training data, but the final submission must adhere to the above restrictions.
- **Machine learning is not required** and is not the focus of this competition.
- **Disqualification:**
  - Code that does not compile will be disqualified.
  - Participants found abusing the Gradescope backtester for unfair advantages will be removed.

7 Recommended Approach to the Case

1. **Training & Testing**:
   - Train your algorithm using the provided training data.
   - Locally test performance before submission.
2. **Submission & Leaderboard**:
   - Submit before the deadline.
   - Gradescope will evaluate performance on the **test dataset** (immediately following the training dataset).
   - Leaderboard will display rankings for individual metrics, but **final scores** will be determined after submissions close.
3. **Revise Model**:
   - Repeat steps 1 and 2 based on information you learned from step 2 until you're satisfied with your algorithm and/or the submission deadline has passed.
4. **Final Evaluation**:
   - The final run of algorithms will take place **before the competition day (March 1st)**.
   - Results will be announced on the day of the competition.

8 Tips for Success

- **Analyze returns, not prices**: Stock prices are typically non-stationary, while returns are more stable.
- **Avoid overfitting**: Do not test your strategy on the same data you train it on.
- **Submit early**: Ensure your code compiles correctly before the deadline. Take advantage of the backtester we provided.

We look forward to seeing your submissions and wish you the best of luck!