## 1. Data Preprocessing

* Exclusion of Non-Tradable Stocks:

The very first step involved filtering out the stocks that could not be traded, ensuring the foundation of the analysis is robust.

* Shift in Date Columns:

With an objective to predict the returns for the next day, columns from d1 to d11 were shifted by a day.

* Feature Engineering:

Trading Volume Rate of Change: This metric provides insights into the liquidity and interest in the stock, which often correlates with its future price movement.

Lagged Features: Past data can often provide insights into future movements. Therefore, trading volumes and returns for the previous five periods were incorporated as features.

Standardization: All the features underwent standardization, ensuring the model doesn't get biased due to variable scales.

Handling Missing Data: Features with a missing data rate of more than 50% were removed. This helps in keeping the dataset rich in information while minimizing noise.

## Dataset Splitting

For each stock:

Training Set: The initial 80% of the time series data.

Validation Set: The subsequent 10%, i.e., from 80% to 90%. This set is crucial for hyperparameter tuning.

Backtesting Set: The final 10% of the data, ensuring the model's robustness and its applicability to unseen data.

## Model Selection: XGBoost

Given the presence of missing values, especially for di , a deep learning model might not be the most suitable. Tree-based models like XGBoost can inherently handle missing data, making them a good fit. While linear regression models can be used for such tasks, their learning capabilities are often overshadowed by powerful algorithms like XGBoost.

## Portfolio Construction

Synthetic Factor: The output of the XGBoost model was viewed as a newly synthesized factor.

Smoothing: A 5-day smoothing was applied to this factor, ensuring that short-term noise in the predictions was minimized.

Daily Trading Strategy:

On each day in the test set, stocks were ranked based on the value of this factor.

Long: Top 10% of stocks.

Short: Bottom 10% of stocks.

Equal Weighting: To ensure diversification and manage risks, an equal-weighted portfolio was constructed using the selected stocks.

## Strategy Analysis

Upon executing the strategy, various metrics were computed to assess its performance:

Sharpe Ratio: An indicator of the risk-adjusted performance of the portfolio.

Maximum Drawdown: Gives insights into potential losses in adverse scenarios.

Win Rate: Helps in understanding the consistency of the strategy.

Conclusion: By combining data preprocessing, feature engineering, a robust modeling approach with XGBoost, and a systematic trading strategy, we aim to navigate the complexities of the stock market and derive meaningful insights and returns. However, it's crucial to periodically review and adjust the strategy as markets evolve.

## Portfolio Performance

In summary, with daily rebalancing, the strategy yields a Sharpe ratio of 1.2, a win rate of 54%, and a volatility of 23.9%. However, its maximum drawdown is alarmingly high, reaching 97%. The overall performance is somewhat disappointing. Taking transaction fees into account, I believe it would be more appropriate to reduce the trading frequency, perhaps to a weekly rebalancing. I see potential for improvement in crafting more valuable factors. Given my past experience with high-frequency price-volume data, I'm less familiar with daily data. I made several attempts, but the results were not up to par. Given more time, I would endeavor to construct more effective factors. Regarding the model, I regret not exploring deep learning models. It's plausible to fill in missing values in the features based on group categories, such as using KNN algorithm. However, due to time constraints, I didn't venture into these alternatives."



