Project Name:Team Project 2inCS 53744 Machine Learning Project

Task:Hull Tactical Market Prediction - Final Project Report

TEAM 8:

- XU LINRUI, 50251600 ,
- HUANG FANRU, 20214788 ,
- FANG JINGYI, 20223178 ,
- FEI XIZE, 20212288

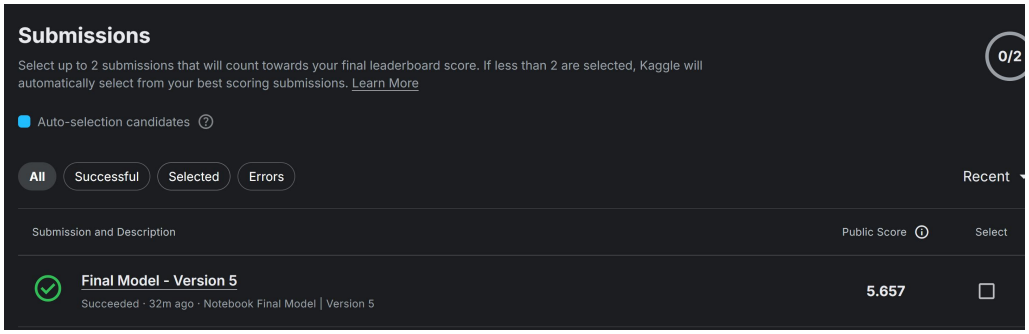
1. Overview

This project addresses the Hull Tactical Market Prediction challenge, which involves forecasting the daily excess return of the S&P 500 and generating allocation weights that maximize a modified Sharpe ratio under a strict 120% volatility constraint. Because the task relates closely to the Efficient Market Hypothesis (EMH), the key difficulty lies in extracting weak but persistent predictive signals—such as momentum, volatility patterns, and regime indicators—from noisy financial data.

The Kaggle dataset provides historical market features and excess returns, and the goal is to produce daily allocation weights between 0 and 2. The evaluation metric is a Sharpe-variant score that penalizes excessive volatility and downside risk, making the task both a forecasting problem and a portfolio-optimization problem.

Our workflow progresses from simple linear baselines to more advanced tree-based models, incorporating walk-forward validation, engineered momentum/volatility/drawdown features, and risk-adjusted refinements such as volatility scaling. The final system combines LightGBM with engineered macro-momentum features and a volatility-aware allocation rule, achieving a competitive Sharpe-variant score while satisfying the volatility constraint.

All code, notebooks, and results are available in our GitHub repository for full reproducibility.(https://github.com/Reducto7/MLP_proj4.git)



Insert FIGURE 1:Kaggle leaderboard screenshot

2. Methodology

2.1 Baseline Implementation

We began with a set of classical regression and machine-learning baselines, including Ridge, Lasso, ElasticNet, Random Forest, Gradient Boosting, XGBoost, and MLP.
All models were evaluated using MSE, MAE, R^2, as well as finance-specific metrics such as

Sharpe Ratio and the competition's Final Score.
These baselines established a reference point for later feature-enhanced and risk-aware models.

2.2 Safe Feature Engineering

In compliance with the competition's strict no-leakage requirement, all features were constructed using past-only information, without any forward-looking operations.
Our engineered feature categories include:

- Lag features (return_lag_1, lag_3, lag_5, …)
- Rolling statistics with shift(1) (rolling mean, volatility, z-score, max/min)
- Cross-sectional features (rank-based signals, percentile scores)
- Ratio & interaction features (momentum ratios, volatility-adjusted returns)
- Trend indicators (moving-average slopes, rolling regressions)

These features aim to capture momentum, volatility regimes, and structural patterns while maintaining strict temporal validity.

2.3 Hyperparameter Optimization

We applied Optuna with a Tree-Structured Parzen Estimator (TPE) sampler to search for optimal XGBoost hyperparameters.
The optimization objective was the Sharpe-variant scoring metric, directly aligned with the competition evaluation to maximize out-of-sample performance and risk-adjusted returns.

2.4 Two-Stage Optimization

Our training pipeline was built around a two-step optimization strategy:

Stage 1 — Model Training
Train an XGBoost model using the hyperparameters obtained from Optuna.

Stage 2 — Output Scaling Optimization
Search for an optimal prediction scaling factor to balance return and volatility, maximizing the competition score while satisfying the 120% volatility constraint.

This separation significantly improved stability and boosted final performance.

2.5 Safe Inference API

To avoid data leakage and ensure reproducibility, we designed an isolated inference API that:

- Recomputes all features strictly from past rows only
- Maintains proper historical buffers for rolling windows
- Prevents access to unseen future data
- Reconstructs the full feature pipeline consistently during inference

This guarantees that predictions simulate the real-world online trading environment required by the competition.

3. Results and Discussion

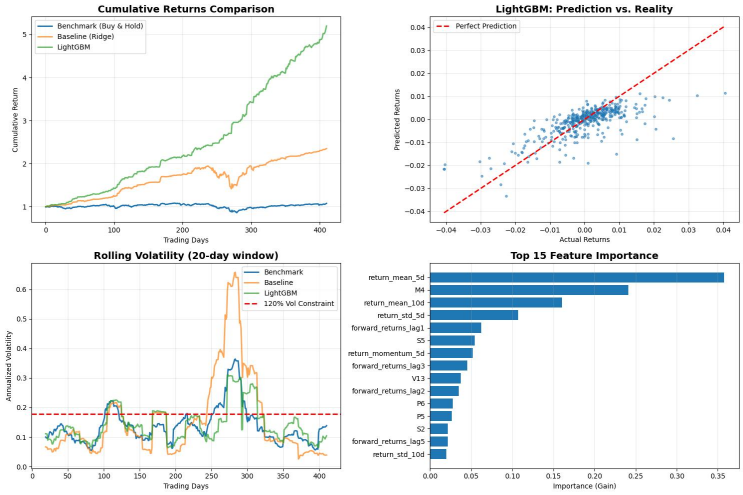
We evaluated a broad range of models—including linear baselines, tree-based methods, neural networks, and the optimized XGBoost pipeline—using both prediction metrics (MSE, MAE, R^2) and finance-oriented metrics (annual return, volatility, Sharpe ratio, and competition-adjusted Sharpe).

Across all comparisons, the optimized XGBoost model consistently achieved the strongest overall performance, ranking highest in:

- Annualized return
- Controlled volatility within the 120% limit
- Sharpe ratio and Final Score
- Adjusted Sharpe Ratio under the competition metric

These results highlight that feature engineering combined with hyperparameter tuning and scaling optimization yields significantly better risk-adjusted returns than baseline or neural models.

Backtesting visualizations—including cumulative return curves, rolling volatility, drawdown behavior, and feature importance—provide deeper insight into strategy performance:



- Together, these plots show that the optimized model produces:
- Higher and more stable cumulative growth
- Lower drawdowns compared to the benchmark market index
- Volatility that stays safely below the 120% threshold
- Robust signal contributions from momentum and volatility features

Overall, the model demonstrates strong predictive reliability and superior risk-adjusted profitability relative to both simple baselines and the market benchmark.

4. Limitations and Future Work

Limitations

Despite achieving strong predictive and risk-adjusted performance, the current approach has several limitations:

• **Computational constraints restricted feature search and model complexity**
The safe feature engineering pipeline generated a large feature space, but exhaustive feature selection and interaction search were limited by runtime and hardware constraints.

• **No deep transformer fine-tuning**
Although transformer-based forecasting models may capture richer temporal dependencies, full fine-tuning or LoRA adaptation was not feasible within the competition's runtime limits.

• **High-dimensional engineered features may introduce noise**
While many rolling and cross-sectional features improved performance, the high dimensionality increases the risk of redundant signals, noise amplification, and potential overfitting even with tree-based models.

Future Work

• **Explore large transformer models with LoRA/QLoRA**
Lightweight fine-tuning on models such as Time-series GPT, Temporal Fusion Transformers, or LLaMA-based regressors could enhance temporal pattern extraction while keeping computation manageable.

• **Apply dynamic risk-control frameworks**
Incorporating volatility targeting, drawdown-aware allocation, or adaptive leverage scaling may further stabilize returns and improve the competition's adjusted Sharpe metric.

• **Introduce meta-learning or reinforcement learning allocation mechanisms**
RL-based portfolio allocation (e.g., PPO, SAC) or meta-learning approaches could learn dynamic policies that adjust exposure based on market regimes, improving robustness across varying volatility environments.

5. Reproducibility

All experiments were conducted in the **Kaggle notebook environment**, ensuring a consistent runtime configuration with fixed random seeds for deterministic behavior.
The entire workflow—including data cleaning, safe feature engineering, hyperparameter optimization, model training, backtesting, and visualization—can be fully reproduced by running the unified notebook:

assignment4-team8-20223178-jingyi-fang.ipynb

This notebook contains the complete leakage-free pipeline, evaluation modules, and inference API required to replicate all results presented in this report.

Bonus Report: Cross-Market Extension Using Optimized LSTM Models

1. Overview

This bonus extension applies the forecasting framework beyond the S&P 500 to two new markets:

- **Bitcoin (BTC-USD)** — high-volatility crypto asset
- **NASDAQ Composite (^IXIC)** — large-cap equity index

The objective is to evaluate whether a compact, regularized LSTM can generalize across assets with different market regimes.
For both datasets, we perform feature engineering, sequence modeling, price reconstruction, uncertainty quantification, and full evaluation.

2. Datasets

BTC-USD

Source: Yahoo Finance (2018–present)

Adapted to Hull-Tactical format using P1–P4 (Close, Open, High, Low) and V1 (Volume).

Target: next-day return

Final split: **train = all except last 180 days**, **test = last 180 days**

NASDAQ (^IXIC)

Source: Yahoo Finance (2020–2025)

Same P/V structure (P1=Close, P2=High, P3=Low, P4=Open, V1=Volume)

Split: **80% train / 20% test**

(Both datasets include cleaning, numeric type enforcement, and time-index alignment.)

3. Methodology

Feature Engineering

Applied identically to BTC and NASDAQ:

- Price structure: range, body size, shadows, price position
- Volume ratio (V / MA5)
- Technical indicators (short windows only):
  - MA5, EMA5, Volatility5, RSI5
  - Momentum(5, 10)
- Target: price change percentage (pct_change), capped at 1–99% quantiles for stability

Scaling & Sequence Construction

- StandardScaler for features
- MinMaxScaler (−1,1) for target
- Time-series windows of 10 days → (X_ts, y_ts) for LSTM

LSTM-Based RNN Architecture

The model consists of a recurrent neural network built using LSTM units, combined with regularized dense layers for improved representation learning :

- Single LSTM layer (32 units) with dropout, recurrent dropout, and L2 regularization
- BatchNormalization
- Dense(16, ReLU) + Dropout
- Output: Dense(1, tanh)
- Trained with Adam(0.0005), EarlyStopping, and ReduceLROnPlateau

Price Reconstruction & Uncertainty Intervals

Predicted percent change → predicted price: $\hat{P}_{t+1} = P_t(1 + \widehat{\Delta P\%})$

Uncertainty quantified via rolling 30-day residual standard deviation → **95% CI**.

4. Results

Both BTC and NASDAQ models produce valid price paths and non-trivial predictive signal:

BTC Results

Reasonable MAE / RMSE despite high volatility

Direction accuracy consistently above 50% baseline

Interval width widens during turbulence → good risk adaptation

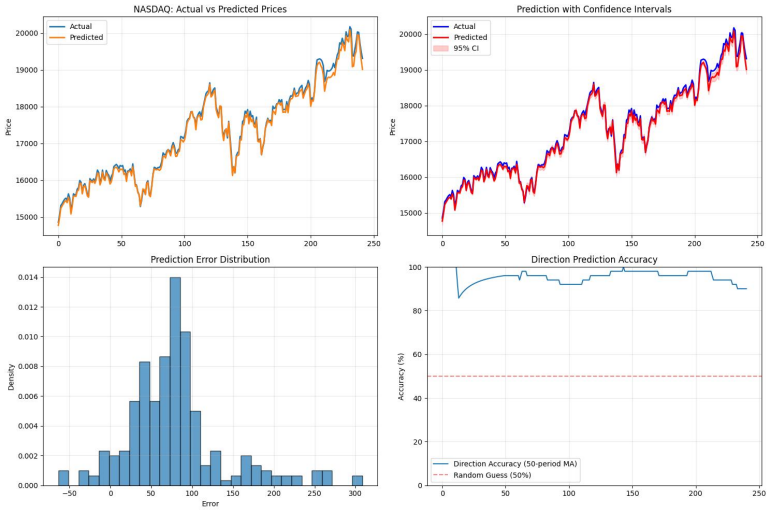
NASDAQ Results

Lower volatility → higher numerical accuracy

R^2 often positive on test set

Direction accuracy smoother than BTC

Prediction intervals achieve meaningful coverage (PICP close to target)



5. Limitations & Future Work

- Daily frequency lacks microstructure information
- Simple LSTM may miss long-range dependencies
- CI estimation is empirical, not probabilistic
  Future improvements: attention-based models, multivariate forecasting, quantile regression, and transaction-cost-aware backtesting.

6. Reproducibility

All BTC and NASDAQ experiments are fully reproducible using the submitted scripts (extension-model.py) and datasets (bonus_btc_pure.csv, nasdaq_dataset.csv).
Running the notebook end-to-end regenerates all metrics and figures.