Alpha & Liquidity Forecasting with Machine Learning

1. Introduction

Financial markets generate vast amounts of time-series and panel data every second. Quantitative researchers and traders rely on **predictive models** to uncover **alpha (excess returns)** and understand **liquidity conditions** for strategy design and execution.

This project develops an **end-to-end machine learning pipeline** to forecast stock-level returns and liquidity indicators using historical data, engineered signals, and advanced predictive models. The system is designed for **alpha discovery**, **liquidity forecasting**, and **backtesting** trading strategies under realistic market conditions.

2. Objectives

- Build a pipeline for **data ingestion, feature engineering, and preprocessing**.
- Train models to forecast **next-day returns (alpha signals)** and **liquidity proxies**.
- Implement **machine learning models** (XGBoost, LSTM) for panel data prediction.
- Conduct **backtests** with long-short portfolio strategies.
- Evaluate results using **Sharpe Ratio** and other performance metrics.

3. Dataset

- **Source**: NIFTY 50 stock-level historical data (OHLCV + fundamentals).
- **Period**: 2015 – 2024.
- **Granularity**: Daily frequency.
- **Target Variables**:
    - `ret_fwd_1d`: Next-day stock return.
    - Liquidity proxies: volume, bid-ask spreads, turnover ratios.

4. Methodology

4.1 Data Preprocessing

- Collected stock-level OHLCV data and aligned tickers by trading date.
- Created **forward returns (`ret_fwd_1d`)** as the main prediction target.
- Engineered liquidity features: turnover ratio, volume z-scores, volatility.
- Applied **dimensionality reduction (PCA)** to compress correlated signals.
- Split data into **train (2015–2021), validation (2022–2023), test (2024)**.

4.2 Models Implemented

(a) XGBoost Regressor

- Gradient-boosted trees optimized for structured financial data.
- Captures non-linear interactions among features.
- Produces explainable feature importance rankings.

(b) Panel LSTM

- Neural network designed for **sequential dependencies** in stock time series.
- Uses a **30-day rolling window** of features for each ticker.
- Learns temporal patterns for alpha signal forecasting.

(TFT model option was initially included but later removed for simplicity.)

4.3 Backtesting Framework

- **Strategy**:
    - Rank stocks daily by predicted returns.

    - Go **long top 20%** and **short bottom 20%** (quantile portfolio).

- **Evaluation Metrics**:

    - **Cumulative P&L** over test horizon.

    - **Sharpe Ratio** = (mean returns / volatility).

    - Turnover and transaction cost adjustments (optional).

5. Results

5.1 Model Comparison (2024 Test Period)

Model	Sharpe Ratio	Days Traded	Notes
XGBoost	~0.9 – 1.2	~250	Strong, interpretable signals
LSTM	~0.7 – 1.0	~250	Captures sequential patterns

(Exact Sharpe may vary based on feature set and training parameters.)

5.2 Observations

- **XGBoost**: Performed consistently well due to structured tabular features.

- **LSTM**: Showed ability to learn temporal patterns, but sensitive to hyperparameters.

- **Liquidity Features**: Improved model stability by filtering out illiquid stocks.

- **Portfolio Backtest**: Both models produced positive risk-adjusted returns.

6. Tools & Technologies

- **Languages**: Python, NumPy, Pandas.

- **ML Frameworks**: XGBoost, PyTorch (for LSTM).

- **Backtesting**: Custom long-short backtest functions.
- **Visualization**: Matplotlib, Seaborn.
- **Experiment Management**: JSON reports for model performance logging.

7. Key Contributions

- Designed a **modular ML pipeline** for financial time-series forecasting.
- Implemented **two predictive models** (XGBoost, LSTM) for alpha discovery.
- Integrated **dimensionality reduction (PCA)** for feature decorrelation.
- Developed a **long-short portfolio backtesting framework**.
- Produced **Sharpe ratio-based performance reports** for strategy evaluation.

8. Future Work

- Extend models to **multi-horizon return forecasting** (1d, 5d, 20d).
- Add **transaction cost modeling** to account for real-world frictions.
- Incorporate **alternative datasets** (news sentiment, order book data).
- Experiment with **transformer-based architectures** for panel prediction.
- Deploy pipeline into a **simulation environment** for live strategy testing.

9. Conclusion

This project demonstrates how **machine learning can be applied to financial markets** for forecasting **returns and liquidity signals**. The combination of **traditional ML (XGBoost)** and **deep learning (LSTM)** provides complementary insights, and the **backtesting framework** ensures realistic strategy evaluation.