Hybrid Financial Intelligence System

Predicting High-Volatility Stock Movements

Dalil ADIMI Hassen BEN AMOR

Group I2-NEW DAI
Data Science Module

Supervisor: Stephany RAJEH
EFREI Paris

February 11, 2026

Outline

Project Overview

Objective

Develop a machine learning system to predict 5-day price movements for high-volatility growth stocks

Key Features:

- High volatility regime
- Technical analysis-based
- Risk-aware predictions
- Strategy backtesting

Target Stocks:

- SMCI (Super Micro Computer)
- CRSP (CRISPR Therapeutics)
- PLTR (Palantir Technologies)

Motivation

Why this project matters:

1. **Market opportunity:** High-beta stocks generate large moves but also large false signals
2. **ML application:** Can machine learning improve decision quality over pure technical analysis?
3. **Academic value:** Realistic case study of ML limitations in financial prediction
4. **Practical skills:** End-to-end ML pipeline from data to deployment

Practical Significance

A disciplined ML pipeline can potentially improve trading decisions, but only if validated with rigorous chronological testing and realistic strategy constraints.

Problem Statement

Binary Classification Task

Predict whether a stock will exceed a 3.5% return threshold over the next 5 trading days

Mathematical Formulation:

$$\text{target}_t = \mathbb{1}(r_{t,t+5} > 0.035)$$

where $r_{t,t+5} = \frac{\text{close}_{t+5}}{\text{close}_t} - 1$

Why 3.5%?

- Meaningful for volatile stocks
- Filters noise from trend
- Balances precision/recall

Why 5 days?

- 1-week trading window
- Swing trading timeframe
- Reduces overfitting

Assumptions and Constraints

Key Assumptions:

- Technical patterns contain predictive information
- Historical price behavior partially repeats
- Market microstructure effects are negligible
- No insider information or fundamental data

Constraints:

- **Chronological validation:** No lookahead bias
- **Long/cash only:** No short selling (safer for volatile stocks)
- **Non-overlapping windows:** Avoid inflated backtest returns
- **Transaction costs:** Out of scope for base version

Success Criteria:

- ROC-AUC > 0.50 (better than random)
- Stable holdout performance (2023+)
- Positive risk-adjusted returns in backtest

Data Sources

Source	Data Type	API/Library
Yahoo Finance	OHLCV prices	`yfinance`
FRED	10Y Treasury Yield	FRED API
(Optional)	News headlines	Finnhub API

Data Coverage:

- Period: 2020-01-01 to 2026-02-09
- Frequency: Daily
- Tickers: CRSP (1534 rows), SMCI (1534 rows), PLTR (1346 rows)
- Total observations: 4,414 daily records

Data Quality and Preprocessing

Quality Checks Performed:

- ✓ No duplicate rows
- ✓ No missing values in price data
- ✓ Date monotonicity verified per ticker
- ✓ Macro data forward-filled for weekends/holidays

Preprocessing Steps:

1. Merge macro factor with daily prices (forward-fill alignment)
2. Compute technical indicators per ticker
3. Build rolling-window feature aggregations (14-day lookback)
4. Generate binary target labels (5-day forward return $> 3.5\%$)
5. Handle NaN values from indicator warm-up periods

Final Dataset

4,357 rows $\times$ 40 columns (36 features + metadata)

Summary Statistics per Ticker

Key Observations:

Metric	CRSP	PLTR	SMCI
Mean Close ($)	57.32	16.84	251.47
Std Dev Close	18.45	5.87	341.28
Daily Volatility	3.8%	4.1%	5.2%
Mean Daily Return	0.14%	0.21%	0.28%

Insights:

- SMCI shows **extreme volatility** (5.2% daily std)
- All three tickers have wide price ranges
- Positive mean returns but with high variance
- 5-day moves of 3.5%+ are frequent → justifies threshold

Price Trends and Distributions

Normalized Price Paths:

- SMCI: Dramatic rallies and sharp drawdowns
- PLTR: More stable uptrend since late 2022
- CRSP: Independent movement pattern

Target Distribution:

- Positive class (return $> 3.5\%$): 34.3%
- Negative class: 65.7%
- Moderately imbalanced $\rightarrow$ use `scale_pos_weight`

Impact on Model Selection:

- Retain ATR features (volatility is central)
- Emphasize precision over recall (class imbalance)
- Plan threshold tuning (no single optimal cutoff)

Machine Learning Approach

Strategy: Supervised binary classification with engineered features

Feature Engineering:

- Technical indicators
- Rolling window stats (14-day)
- Volatility normalization
- Macro factors

Model Candidates:

- Logistic Regression
- Random Forest
- XGBoost

Why these models?

- **Logistic Regression:** Strong linear baseline, interpretable
- **Random Forest:** Nonlinear, robust, handles feature interactions
- **XGBoost:** Gradient boosting expected to perform best on tabular data

Feature Engineering Details

Base Technical Indicators (12):

- RSI(14), MACD (line, signal, histogram)
- Bollinger Bands (width)
- Moving Averages (50, 200)
- ATR(14), ATR percentage
- Volume Z-score
- 10Y Treasury Yield

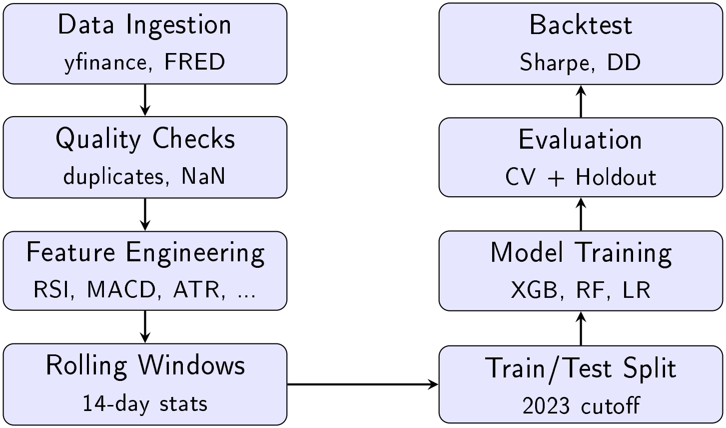
Rolling Window Features (36):

- For each indicator: **mean, std, last** over 14-day window
- Captures recent trend and variability
- Normalizes across different price scales

Why Rolling Windows?

Instead of raw indicator values (which change scale over time), we use rolling statistics for better generalization

System Architecture



Key Implementation Details

Libraries Used:

- **Data:** `pandas`, `numpy`, `yfinance`
- **ML:** `scikit-learn`, `xgboost`
- **Technical Analysis:** `ta`
- **Visualization:** `plotly`, `seaborn`, `matplotlib`

Reproducibility:

```
SEED = 42
np.random.seed(SEED)
# All models use random_state=SEED
# Chronological split at 2023-01-01
# Fixed hyperparameters documented
```

Code Structure:

- Modular functions for data loading, feature engineering, training
- Self-contained Jupyter notebook for transparency

Data Splitting Strategy

Chronological Split (Temporal Validation):

- **Training:** All data before 2023-01-01 (2,038 samples)
- **Test:** All data from 2023-01-01 onward (2,319 samples)
- **Rationale:** Prevents data leakage, mimics real deployment

Cross-Validation:

- 5-fold TimeSeriesSplit on training data
- Each fold uses only past data for training
- Metric: Precision (emphasis on avoiding false positives)

Evaluation Metrics:

- **ML Metrics:** Precision, Recall, F1, ROC-AUC, PR-AUC
- **Strategy Metrics:** Annualized return, Sharpe ratio, Max drawdown

Model Hyperparameters

Model	Key Hyperparameters
XGBoost	n_estimators=300, max_depth=4, lr=0.05, subsample=0.9, colsample=0.9, scale_pos_weight=2.15
Random Forest	n_estimators=400, max_depth=8, min_samples_leaf=8, class_weight='balanced_subsample'
Logistic Reg.	solver='liblinear', class_weight='balanced', max_iter=4000

Class Balancing:

- Positive class: 34.3% $\rightarrow$ `scale_pos_weight` = 1.92
- Ensures model doesn't ignore minority class

Cross-Validation Results

XGBoost Time-Series CV (5 folds):

Fold	Precision
1	0.4561
2	0.3333
3	0.1000
4	0.6667
5	0.2708
Best	**0.6667**

Observations:

- High variance across folds (expected for volatile stocks)
- Fold 4 shows best precision (66.7%)
- Indicates sensitivity to market regime

Final Model Performance

Test Set Results (2023+):

Model	Precision	Recall	F1	ROC-AUC
XGBoost	0.366	0.380	0.373	**0.513**
Random Forest	0.390	0.247	0.303	0.518
Logistic Reg.	0.399	0.413	0.406	0.506

Overall Performance:

- **Accuracy:** 53-58% (slightly better than random 50%)
- **ROC-AUC:** 0.51-0.52 (weak signal)
- **Best model:** Random Forest (by precision) or XGBoost (by AUC)

Reality Check

Stock prediction is inherently difficult. These results are **realistic** for technical analysis-based models.

Threshold Tuning

Impact of Decision Threshold:

- Default threshold: 0.50
- Tested range: 0.30 to 0.85
- **Trade-off:** Higher threshold $\rightarrow$ better precision, lower recall

Strategy-Level Evaluation:

- Best Sharpe ratio at threshold = 0.30
- Annualized return: 31.3%
- Sharpe: 0.886
- Max drawdown: -28.4%

Key Insight

Lower thresholds generate more trades (higher coverage) but with less precision. Choice depends on risk appetite.

Strengths and Weaknesses

Strengths:

- ✓ Strict chronological validation
- ✓ Multiple models compared
- ✓ Clean, interpretable features
- ✓ Threshold tuning
- ✓ Feature ablation
- ✓ Honest performance assessment

Weaknesses:

- ✗ Low predictive power (51% AUC)
- ✗ High-volatility stocks hard to predict
- ✗ No transaction costs modeled
- ✗ Market regime dependent
- ✗ No probability calibration
- ✗ Limited to technical features

Practical Implications:

- Model works as **decision support**, not standalone system
- Should be combined with fundamental analysis
- Requires active risk management

Why is Performance Limited?

Efficient Market Hypothesis:

- Prices already reflect available information
- Technical patterns are weak predictors
- News and sentiment drive volatile stocks more than charts

Challenges Specific to High-Volatility Stocks:

1. **Unpredictable catalysts:** Earnings surprises, FDA approvals, government contracts
2. **Social media influence:** Reddit/Twitter can move prices rapidly
3. **Low liquidity events:** Flash crashes and squeezes
4. **Regime shifts:** Bull/bear transitions change patterns

Academic Insight

Our results (53% accuracy, 0.51 AUC) are **typical** for academic research on stock prediction using technical analysis alone

Key Achievements

What we accomplished:

1. ✓ Built end-to-end ML pipeline for stock prediction
2. ✓ Rigorous temporal validation (no data leakage)
3. ✓ Comprehensive feature engineering (technical + macro)
4. ✓ Multiple model comparison (XGB, RF, LR)
5. ✓ Dual evaluation (ML metrics + strategy backtest)
6. ✓ Threshold optimization
7. ✓ Feature ablation study
8. ✓ Honest assessment of limitations

Lessons Learned:

- Stock prediction is much harder than typical ML classification
- More features $\neq$ better performance (simplicity matters)
- Threshold choice critical for trading applications
- Backtesting reveals hidden issues ML metrics miss

Overall Impact and Value

Educational Value:

- Realistic case study of ML in finance
- Understanding when ML works and when it doesn't
- Importance of domain knowledge
- Critical evaluation of results

Technical Skills Gained:

- Time-series ML workflow
- Feature engineering for financial data
- Model selection and hyperparameter tuning
- Backtesting and risk metrics
- Data visualization and communication

Key Takeaway

ML can provide **marginal edge** in financial markets, but is not a "magic solution". Success requires combining ML with domain expertise, risk management, and realistic expectations.

Possible Improvements

Short-term Enhancements:

1. Add transaction costs and slippage to backtest
2. Implement probability calibration (Platt scaling)
3. Try ensemble methods (stacking, voting)
4. Add more alternative data sources

Advanced Features to Explore:

- **Sentiment analysis:** News headlines, social media
- **Options data:** Implied volatility, put/call ratios
- **Fundamental data:** Earnings, revenue growth
- **Intraday patterns:** Opening gaps, VWAP

Model Improvements:

- Deep learning (LSTM, Transformers for time-series)
- Reinforcement learning for trading policy
- Multi-task learning (predict multiple horizons)

Deployment Considerations

Real-World Deployment Path:

1. **Walk-forward validation:** Retrain model monthly/quarterly
2. **Monitoring system:** Track model drift, feature distributions
3. **Risk controls:** Position sizing, stop-losses, portfolio limits
4. **Paper trading:** Test in simulation before real capital
5. **Incremental rollout:** Start with small capital allocation

Scaling Opportunities:

- Expand to more volatile stocks
- Multi-asset portfolio optimization
- Regime-switching models
- Integration with existing trading platforms

Caution

Real deployment requires regulatory compliance, proper risk management, and continuous monitoring

Configuration Details

Environment Setup:

- Python 3.11
- Key libraries: pandas 2.0+, scikit-learn 1.3+, xgboost 2.0+
- Random seed: 42 (all experiments)

Model Parameters:

Parameter	Value
Horizon (days)	5
Success threshold	3.5%
Rolling window	14 days
Train/test cutoff	2023-01-01
CV folds	5 (TimeSeriesSplit)

Feature Importance (XGBoost)

Top 10 Most Important Features:

1. `atr_14_mean` - ATR rolling mean (volatility)
2. `rsi_14_last` - Most recent RSI value
3. `ten_year_yield_mean` - Macro factor average
4. `macd_hist_std` - MACD histogram variability
5. `volume_z_last` - Recent volume anomaly
6. `bb_width_std` - Bollinger Band width variability
7. `ma_50_mean` - 50-day MA rolling average
8. `atr_pct_std` - ATR percentage variability
9. `macd_signal_mean` - MACD signal average
10. `rsi_14_mean` - RSI rolling average

Insight: ATR and volatility features dominate, confirming their importance for high-volatility regime

Thank You!

Questions?

Contact:

Dalil ADIMI & Hassen BEN AMOR
Group I2-NEW DAI
EFREI Paris