Forecasting Market Volatility

A Two-Tiered LSTM Alert System for High-Precision Trading Signals

Mohamed Nabil Chagou
YE Suitao

Master 1 DAC
2025 – 2026

Abstract

Short-term directional prediction of stock prices is notoriously difficult, often yielding accuracy no better than a random guess. Confronted with the failure of initial direction-forecasting models (<53% accuracy), this project makes a strategic pivot. We reframe the core task as detecting significant price movements (±2.5%) within a two-day horizon, a more actionable objective for risk management.

Using an LSTM model and a threshold-optimization strategy that prioritizes high-confidence signals, we successfully forecast both the presence and the direction (big rise vs. big drop) of such volatility events. The model demonstrates that high-precision alerts for impending volatility are achievable and hold practical trading value. However, a stark trade-off emerges: while predicting the direction of a move can reach even higher precision, it results in drastically fewer actionable signals, clearly highlighting the increased challenge of directional forecasting compared to simple volatility detection.

Our work illustrates that by pragmatically reframing the forecasting problem—from predicting noisy daily returns to detecting significant volatility regimes—and by optimizing for decision-centric metrics like precision, machine learning can provide structured, valuable guidance for investment decision-making.

Table of Contents

Forecasting Market Volatility

1. Introduction

1.1. Background

Quantitative finance relies on predictive models to identify statistical edges in highly competitive and efficient markets. Among the most sought-after yet challenging goals is **short-term stock price forecasting**. Traditional approaches, ranging from time-series econometrics to machine learning, often aim to predict the next day’ s price direction or return. However, the daily price movement of liquid equities is widely regarded as approximating a random walk, dominated by noise and influenced by a complex mix of macroeconomic announcements, firm-specific news, and market microstructure effects. For quantitative traders, this environment makes consistent profitability through directional prediction exceptionally difficult.

The proliferation of artificial intelligence, particularly deep learning, has introduced new paradigms for navigating this chaos. Models such as recurrent neural networks (RNNs) and transformers offer the capacity to discern complex, non-linear patterns and long-range dependencies within sequential data that traditional linear models may miss. In finance, this translates to the potential for learning subtle, recurring market signatures that precede systematic changes in volatility or liquidity, thereby imposing a form of **statistical order on apparent randomness**. While not a crystal ball, these AI models serve as powerful filters, separating potentially actionable signals from the overwhelming noise of the market. This project is situated within this endeavor, leveraging a deep learning architecture to explore the boundaries of predictability in short-term market dynamics.

1.2. Problematique: From Direction to Volatility

This project began with the conventional objective of building a model to forecast the next-day directional movement (up or down) of a stock. Initial experiments with baseline models confirmed the prevailing wisdom: achieving statistically significant accuracy above a random guess proved unattainable within our framework. This impasse prompted a fundamental reassessment.

Instead of persisting with the elusive task of *directional* prediction, we reframed the problem around a more actionable and potentially modelable market phenomenon: **short-term volatility clustering**. Significant price movements (e.g., $> \pm 2.5\%$), while difficult to assign a direction in advance, often exhibit identifiable precursors in terms of

market tension, volume anomalies, and shifts in volatility regimes. Detecting these precursors can be invaluable for risk management, option positioning, and structuring tactical trades. Consequently, our research question evolved from *"Will the price go up or down tomorrow?"* to ***"Will a significant price movement occur within the next two days, and if so, can we anticipate its direction?"***

1.3. Project Scope and Contributions

This report details the development and evaluation of a forecasting system designed to answer this refined question. Our work makes the following key contributions:

1. **Problem Reformulation:** We define clear binary (volatility presence) and ternary (volatility direction) forecasting targets based on a two-day forward window and a 2.5% movement threshold, creating a structured prediction task grounded in trading utility.
2. **A Precision-First Methodology:** We implement a Long Short-Term Memory (LSTM) network not as a black-box classifier, but as a probability estimator. Its core innovation lies in a subsequent **threshold optimization stage**, where decision boundaries are explicitly tuned to maximize precision, deliberately trading off recall to generate high-confidence alerts.
3. **Practical Evaluation:** We evaluate model performance not only through standard metrics (precision, recall, F1) but also by simulating a simple trading strategy to demonstrate the potential economic value of the generated signals.
4. **Comparative Insight:** By applying the same modeling framework to both binary (presence) and ternary (direction) tasks, we provide a clear empirical comparison of the attainability and trade-offs involved in forecasting volatility versus forecasting direction.

The report is structured as follows: Chapter 2 describes the data and the construction of our forecasting targets. Chapter 3 outlines the LSTM architecture and our threshold-optimization methodology. Chapters 4 and 5 present the results for the binary and ternary forecasting tasks, respectively. Chapter 6 discusses the implications of our findings, and Chapter 7 concludes.

2. Data Acquisition and Initial Analysis

2.1. Data Sources and Collection Pipeline

All market data for this project was programmatically sourced from Yahoo Finance via the official yfinance Python API. We constructed a diversified portfolio of 53 instruments: 50 large-cap U.S. stocks balanced across key sectors (Technology, Finance, Healthcare, Consumer, Energy, Industrials) and 3 major indices (S&P 500, NASDAQ, Dow Jones). Daily OHLCV (Open, High, Low, Close, Volume) data, adjusted for corporate actions, was collected for the ten-year period from January 1, 2015, to December 31, 2024, resulting in an initial dataset of 133,169 records.

2.2. Data Quality Assessment

1. **Missing Values**: A comprehensive check confirmed no missing values in the core price and volume fields across all instruments, indicating excellent data completeness from the source.
2. **Noise and Outlier Detection:** Financial markets are inherently noisy. To quantify this and screen for potential data errors, we calculated the Z-score of daily returns for each instrument using a 30-day rolling window. Figure 2.1 shows the distribution of high-volatility points (Z-score > 3) and extreme outliers (Z-score > 6) across the portfolio.

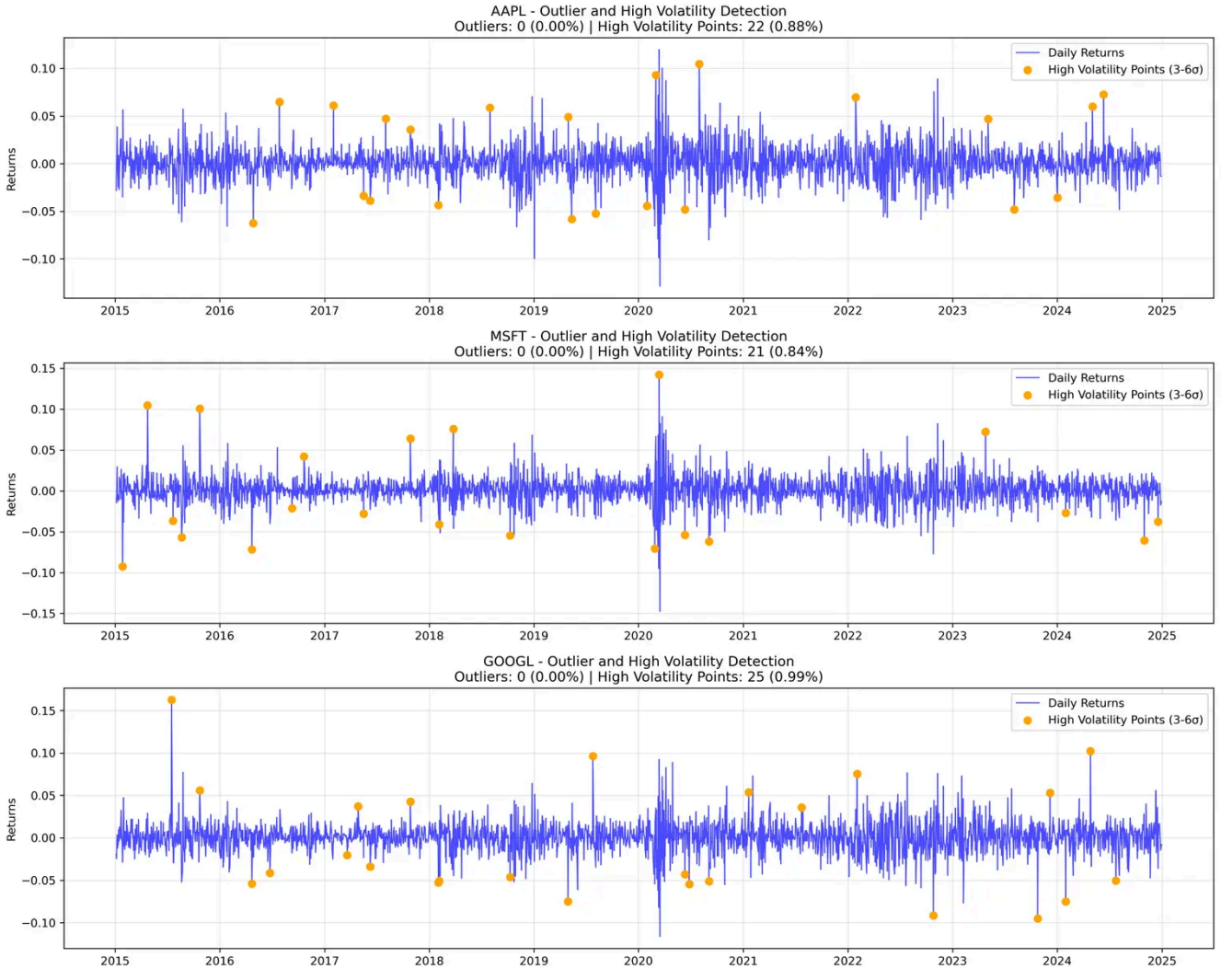


Figure 2.1: Distribution of High-Volatility Points and Extreme Outliers

(Placeholder for your chart of Z-score >3 and >6 distribution)

The analysis revealed no extreme outliers (Z-score > 6), confirming the absence of gross data errors. However, all instruments exhibited a baseline level of high-volatility points (Z-score > 3), ranging from 0.56% to 1.47% of observations. These points correspond to legitimate market events (e.g., earnings announcements, macroeconomic shocks) and were retained as essential components of the market's dynamics.

This pervasive presence of volatility clusters provided a direct motivation and validation for our project’ s core objective: to systematically detect such significant market movements.

3. **Initial Data Characteristics:** The raw dataset was well-structured and required minimal correction. The primary issue identified was a single instance of zero trading volume for the S&P 500 index, which was subsequently handled during preprocessing. The overall quality was high, providing a solid basis for the subsequent feature engineering and modeling stages.

3. Preprocessing

3.1. Forecasting Target Definition

The predictive objective of this project was explicitly designed for practical trading utility. We define a **significant volatility event** as a daily price change exceeding **±2.5%**. This threshold captures moves larger than typical market noise and is economically meaningful for positioning and risk management. Two specific forecasting tasks were derived from this definition.

1. **Binary Classification: Volatility Presence**

The primary task is to forecast *whether* a significant move will occur within the next two trading days. Formally, for a day t, the binary target y_binary is:

y_binary(t) = 1 if max(|return(t+1)|, |return(t+2)|) > 0.025, otherwise 0. where return(t+k) = (Close(t+k) - Close(t)) / Close(t) .

This formulation ensures a forward-looking, two-day warning window.

2. **Ternary Classification: Volatility Direction**

An extended task provides directional granularity. The ternary target y_ternary is:

y_ternary(t) = 0 (Calm) if max(|return(t+1)|, |return(t+2)|) <= 0.025;

= 1 (Big Rise) if max(return(t+1), return(t+2)) > 0.025;

= 2 (Big Drop) if min(return(t+1), return(t+2)) < -0.025.

Cases where both a large rise and drop occur within the window (e.g., high volatility) are assigned the label of the move with the larger absolute magnitude. The resulting label distribution is naturally imbalanced (Calm: ~75%, Big Rise: ~13%, Big Drop: ~12%), directly informing our modeling strategy.

3.2. Feature Engineering

To enable the model to identify precursors to volatility, we constructed an 8-dimensional feature set for each daily observation, combining raw market data with domain-inspired indicators.

Base Features (5): The raw daily Open, High, Low, Close prices and the natural logarithm of trading volume, Volume_log = log(1 + Volume). The log transform stabilizes the variance of the heavily right-skewed volume data.

Engineered Financial Features (3): These features, calculated per instrument with strict temporal causality, encapsulate key market microstructures hypothesized to signal changing regimes.

1. **volatility_ratio_5_20**: The ratio of short-term (5-day) to medium-term (20-day) rolling volatility of returns. Values > 1 suggest accelerating volatility, a potential precursor to large price moves.
2. **momentum_5_sign_strength**: The 5-day return divided by its 5-day volatility. This measures the risk-adjusted momentum strength, filtering out weak or noisy trends.
3. **volume_price_divergence**: The product of 5-day price momentum and 5-day volume momentum. It captures the alignment or divergence between price movement and trading activity—a classic technical analysis concept where divergence often precedes reversals or volatility expansions.

These eight features (Open, High, Low, Close, Volume_log, volatility_ratio_5_20, momentum_5_sign_strength, volume_price_divergence) form the complete input vector for the model.

3.3. Final Preprocessing and Sequential Dataset Creation

The final step transformed the tabular time series into a format suitable for the LSTM model.

1. **Cleaning and Normalization**

The zero-volume record identified in Section 2.2.3 was removed. All features were then **standardized** (scaled to zero mean and unit variance). Critically, the scaling parameters (mean and standard deviation) were calculated **only on the training set** and then applied to the validation and test sets to prevent any data leakage.

2. **Sequence Generation for LSTM**

To capture temporal dependencies, the continuous time series for each instrument was segmented into **sliding windows of 30 consecutive days**. Each window forms a 30×8 matrix (30 timesteps, 8 features). The label for this sequence is the target (y_binary or y_ternary) of the **day immediately following the window** (i.e., day 31). This process generated **128,698 valid sequences**.

3. **Chronological Train-Validation-Test Split**

The sequences were split **in chronological order** to preserve the time-series integrity and simulate a realistic backtest:

- **Test Set:** The last 20% of sequences (25,740 samples) were held out for final evaluation.
- **Training & Validation:** The first 80% of sequences (102,958 samples) were further split temporally: 80% for training (82,366 samples) and 20% for validation (20,592 samples). The validation set was used exclusively for hyperparameter tuning and the crucial **threshold optimization** described in Chapter 4.

This structured pipeline resulted in clean, normalized, and chronologically partitioned 3D datasets ready for model training: (samples, timesteps=30, features=8).

4. Methodology: An LSTM Framework Optimized for Precision

4.1. Model Architecture: Long Short-Term Memory Network

At the core of our forecasting system is a **Long Short-Term Memory (LSTM)** recurrent neural network. LSTM was chosen for its proven ability to capture complex, long-range temporal dependencies in sequential data—a critical requirement for modeling financial time series where patterns may unfold over weeks and interact with historical context.

The network was implemented in TensorFlow/Keras with the following architecture:

- **Input Layer:** Accepts sequences of shape (30, 8), corresponding to 30 days of historical data and 8 features per day.
- **LSTM Layer:** A single LSTM layer with 64 units. This layer processes the sequential input, learning to retain relevant information and forget noise over the 30-day window.
- **Regularization:** A Dropout layer (rate=0.3) follows the LSTM to prevent overfitting, immediately succeeded by a Batch Normalization layer to stabilize and accelerate training.

- **Dense Layers:** Two fully connected (Dense) layers (32 and 16 units, with ReLU activation) further transform the learned representations. Another Dropout layer (rate=0.2) is applied after the first Dense layer.
- **Output Layer:**
  - For the **binary** task: A single neuron with a sigmoid activation, outputting a probability between 0 and 1 for the "significant volatility" class.
  - For the **ternary** task: Three neurons with a softmax activation, outputting a probability distribution over the three classes (Calm, Big Rise, Big Drop).

The model was compiled using the Adam optimizer with a learning rate of 0.0005. The binary model used binary_crossentropyas its loss function, while the ternary model used categorical_crossentropy. Crucially, given the severe class imbalance, **class weights** were computed inversely proportional to class frequencies and supplied during training to penalize misclassification of the minority classes more heavily.

4.2. Core Innovation: Threshold Optimization for Actionable Signals

The conventional approach of using a fixed 0.5 decision threshold fails in imbalanced, high-stakes domains like finance. Our core contribution is a **two-stage process**: first, the LSTM learns to output well-calibrated probabilities; second, we perform a systematic **threshold optimization** to convert these probabilities into high-confidence trading alerts. This decouples pattern learning from risk management, allowing explicit control over the precision-recall trade-off.

4.3. Binary Classification: Single-Threshold Optimization

For the binary task (volatility presence), the model outputs a single probability **p**. The optimization workflow is as follows:

1. **Probability Generation:** The trained LSTM predicts probabilities for all samples in the validation set.
2. **Grid Search:** We evaluate a fine-grained range of candidate thresholds **T** (e.g., from 0.5 to 0.95).
3. **Performance Mapping:** For each **T**, we calculate key metrics: Precision, Recall, and the number of generated signals (where **p > T**).
4. **Visual Analysis & Selection:** This process generates two critical diagnostic plots:

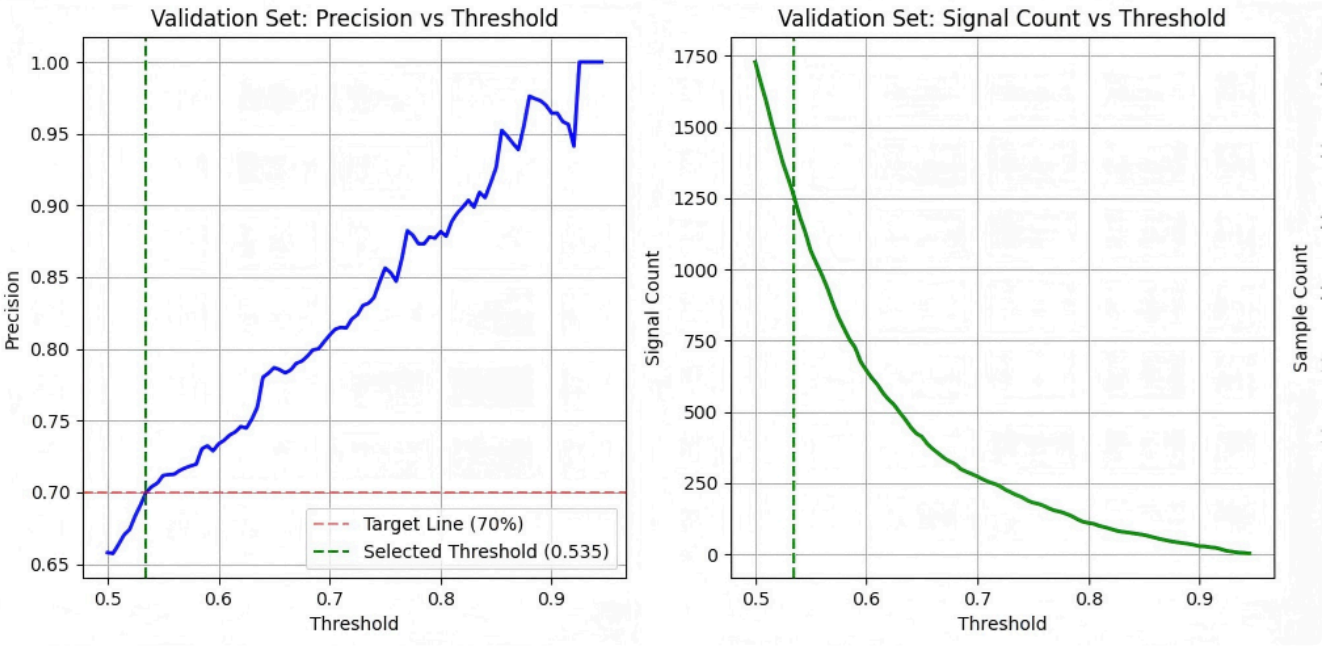


Figure 4.3.1 (Precision vs. Threshold) (Signal Count vs. Threshold): Shows how precision for the volatility class increases monotonically as the threshold rises and corresponding exponential decay in the number of actionable alerts.

5. **Decision Rule:** We define a **target precision** (e.g., ⩾70%) based on the minimum confidence required for a trade. The optimal threshold **T_binary*** is selected as **the lowest threshold that meets or exceeds this target precision**. If multiple thresholds satisfy the condition, we choose the one yielding the **highest signal count** to maximize utility. This point is clearly marked on **Figures 4.3.1**.

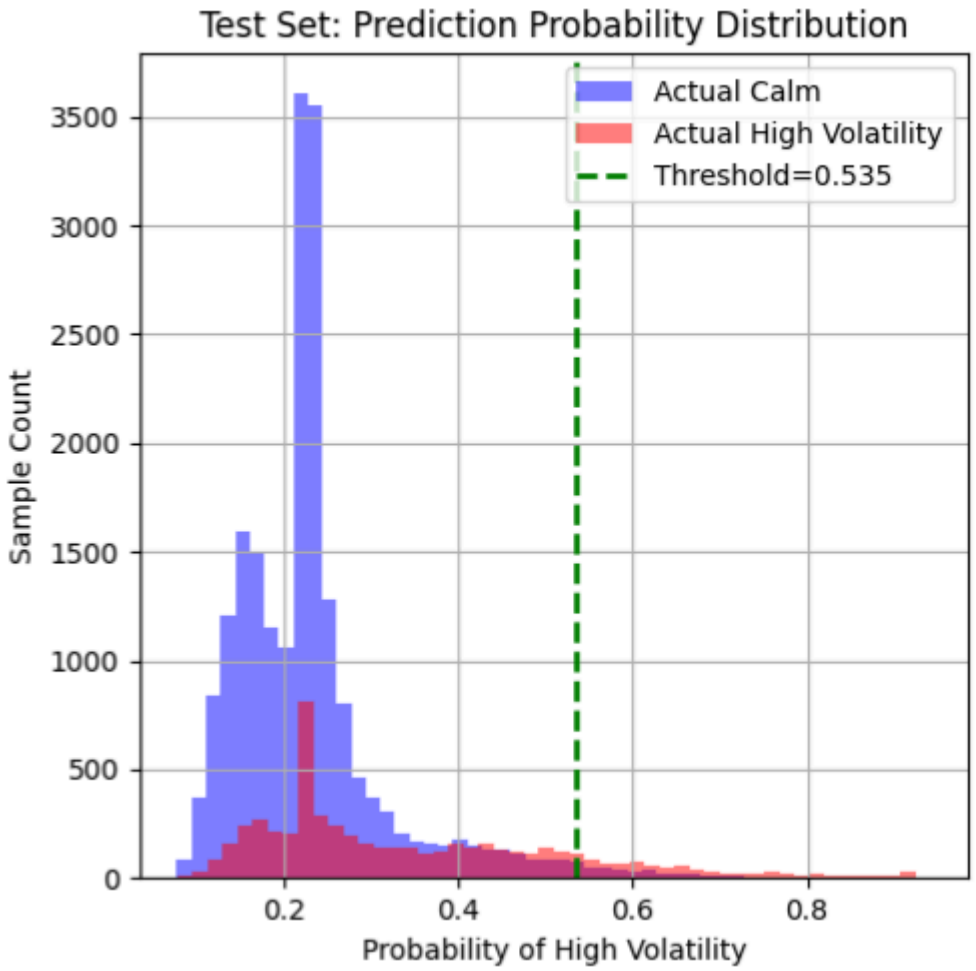
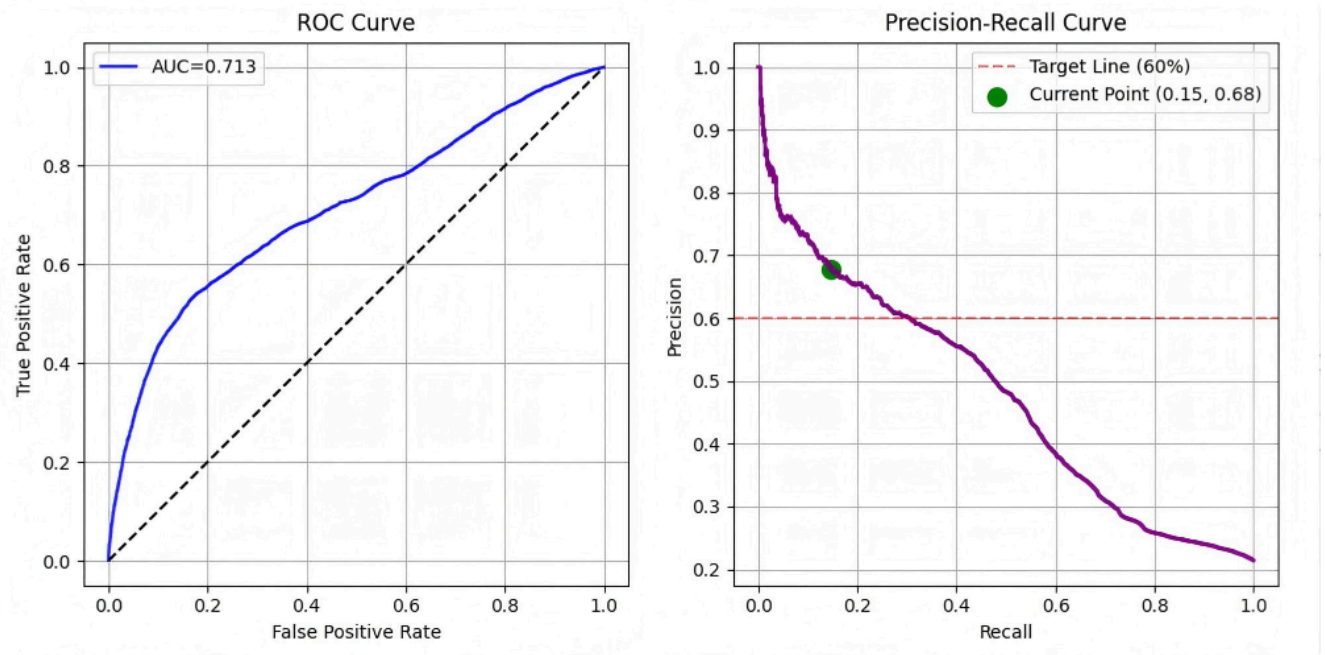


Figure 4.3.2: Distribution of Predicted Probabilities on Test Set.

The model's output probabilities for actual calm days (blue) are concentrated near zero, while probabilities for high-volatility days (red) show a rightward shift, indicating discriminative power. The overlap explains the precision-recall trade-off.



- **Figure 4.3.3: Receiver Operating Characteristic (ROC) Curve.**

The model's ROC curve (AUC = 0.713) lies consistently above the random classifier line (black dashed), confirming its ability to distinguish between calm and high-volatility regimes better than chance.

- **Figure 4.3.4: Precision-Recall Curve with Operating Point.**

The Precision-Recall curve for the minority class (high-volatility). The green dot marks the system’s operating point (Precision=0.68, Recall=0.15), highlighting the high-precision, low-recall strategy essential for minimizing false trading signals.

This methodology ensures every alert has a verified high probability of being correct, directly minimizing costly false positives in trading.

4.4. Ternary Classification: Dual-Threshold Optimization

Predicting direction (Big Rise/Big Drop) introduces a more complex trade-off space. The model outputs a probability vector [p_calm, p_rise, p_drop]. We introduce a dual-threshold rule:

- **Predict “Big Rise” if (p_rise > T_rise) AND (p_rise == max(p_calm, p_rise, p_drop))**
- **Predict “Big Drop” if (p_drop > T_drop) AND (p_drop == max(p_calm, p_rise, p_drop))**
- **Otherwise, predict “Calm”.**

The optimization for T_rise and T_drop is a two-dimensional search conducted on the validation set, with results comprehensively visualized to guide the final decision.

1.2D Grid Search & Performance Surface

We independently vary T_rise and T_drop over a grid (0.3 to 0.8). For each pair (T_rise, T_drop), we calculate the precision for the “Big Rise” class, the “Big Drop” class, and the total number of directional signals generated.

2.Visualization & Diagnostic Analysis

The optimization landscape and model behavior are analyzed through a multi-faceted diagnostic panel (Figure 4.4.1 - 4.4.10), which serves three key purposes: understanding the *optimization trade-offs*, diagnosing the *model's probabilistic calibration*, and validating the *quality of the selected signals*.

2.1. Model Calibration and Probability Diagnostics

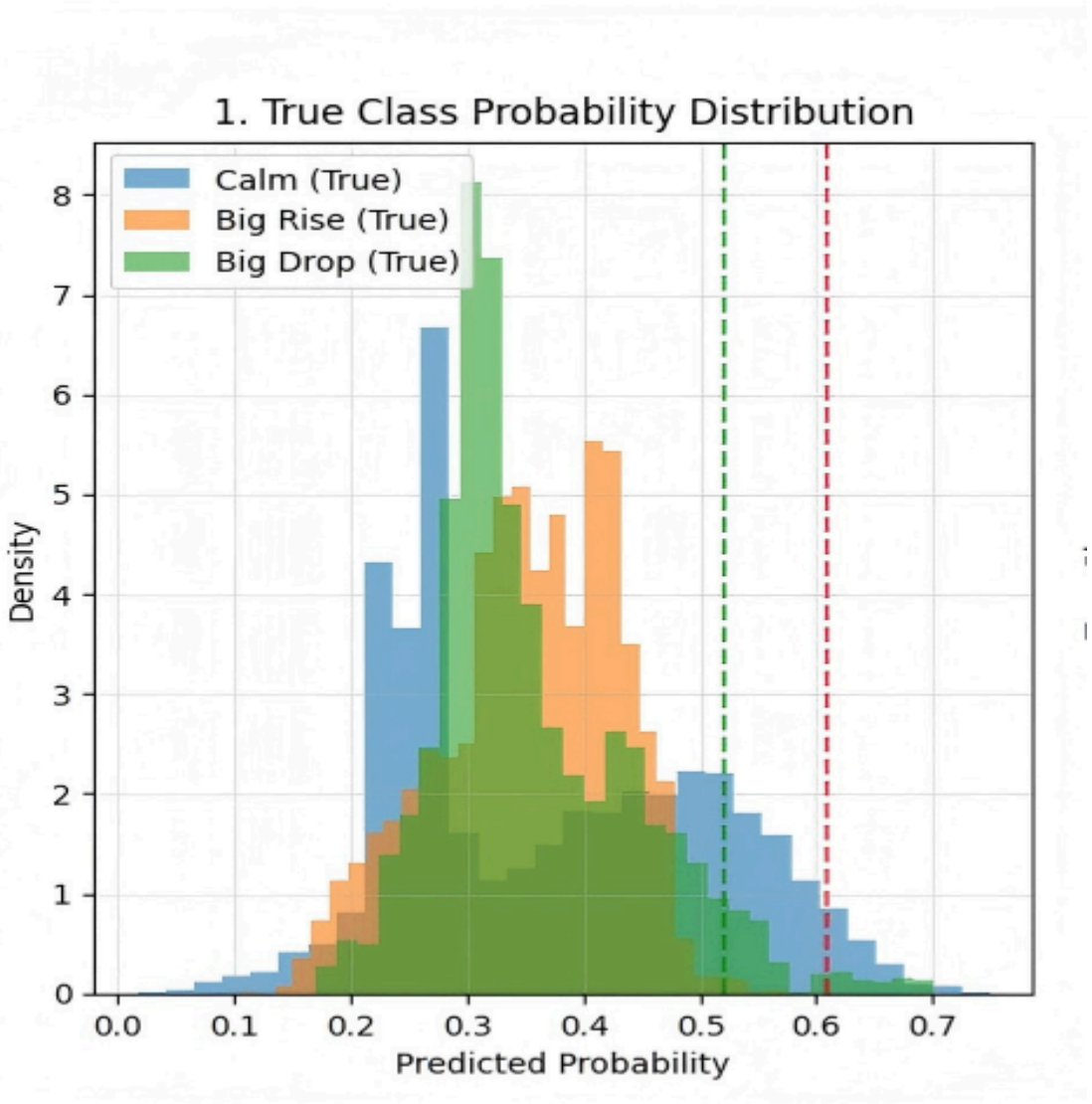


Figure 4.4.1 (Probability Distribution by True Class): Histograms showing the model's output probabilities for samples where the true label is Calm, Rise, or Drop, indicating calibration quality.

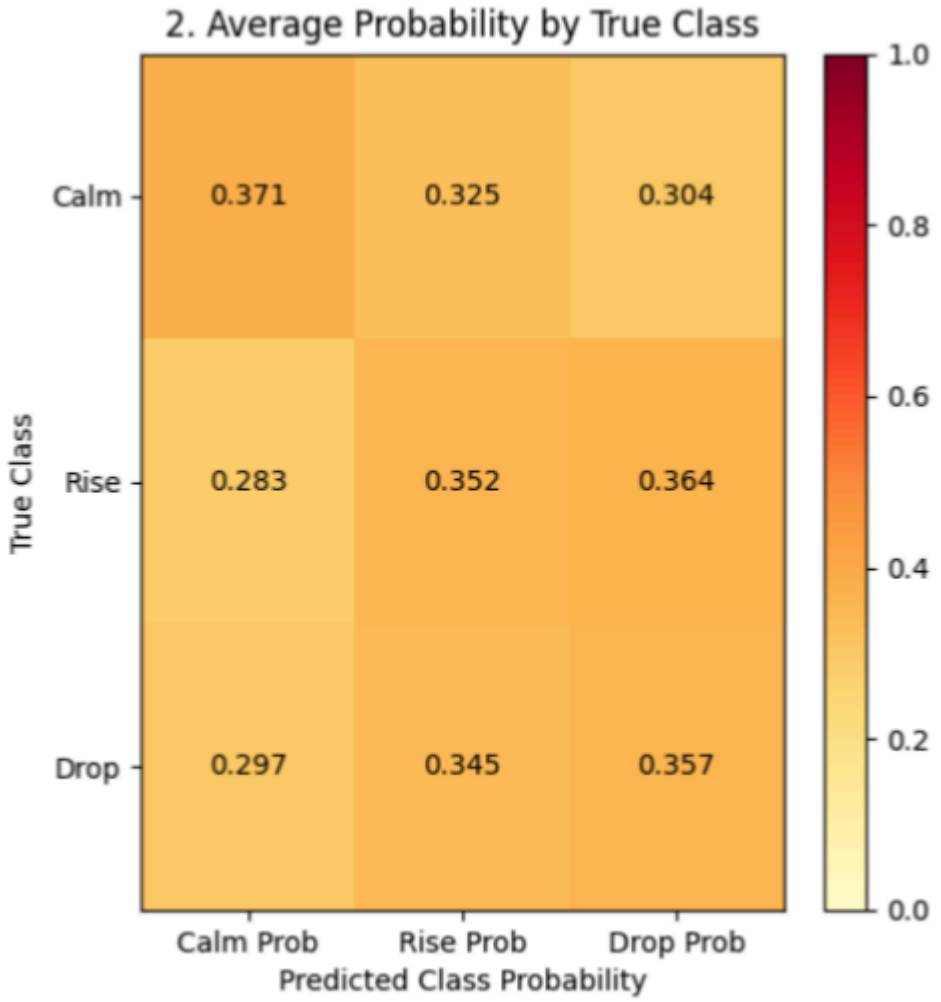
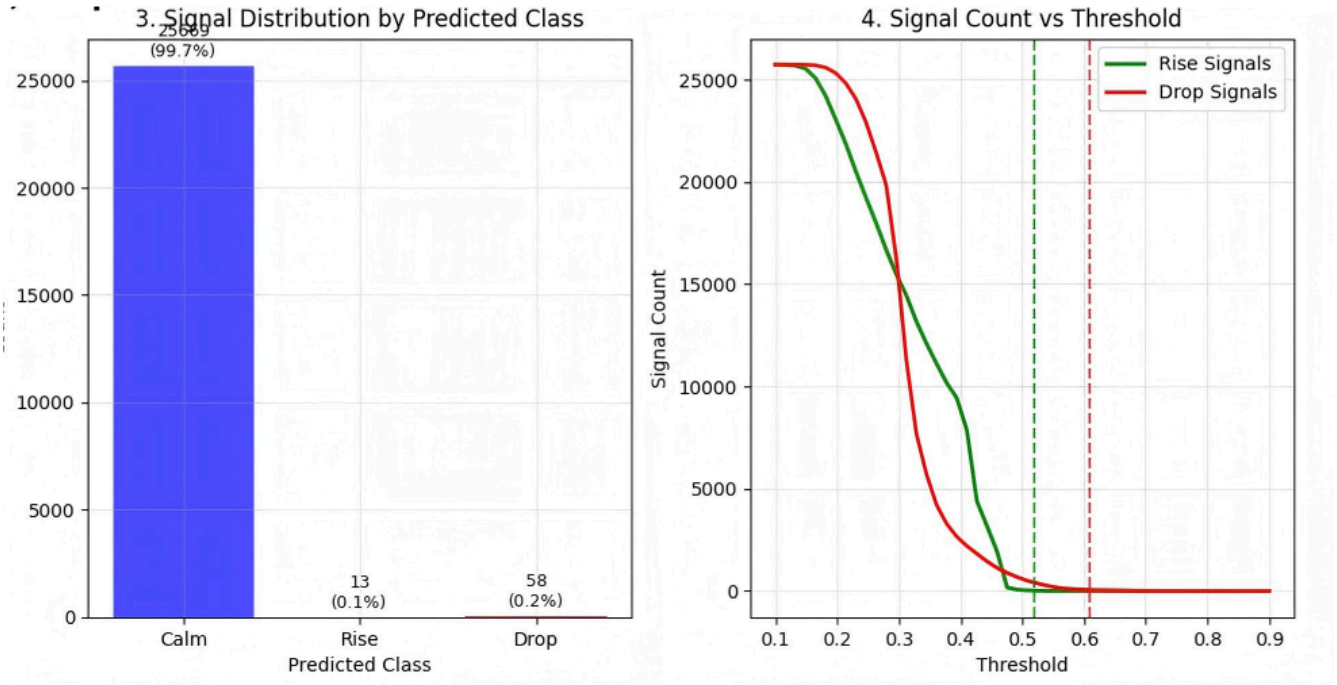


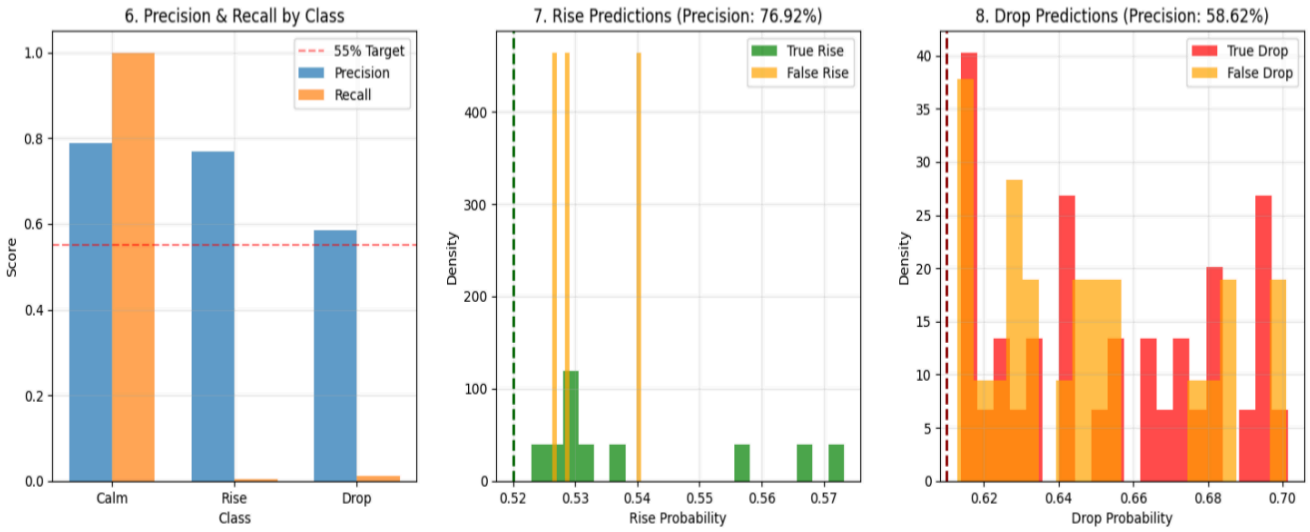
Figure 4.4.2 (Average Predicted Probability Matrix by True Class): A 3×3 heatmap summarizing the mean probability assigned to each predicted class, conditioned on the true class.



- **Figure 4.4.3 (Distribution of Generated Signals by Predicted Class):** A bar chart showing the count of final predictions categorized as Calm, Rise, or Drop, illustrating the strategy's output frequency.

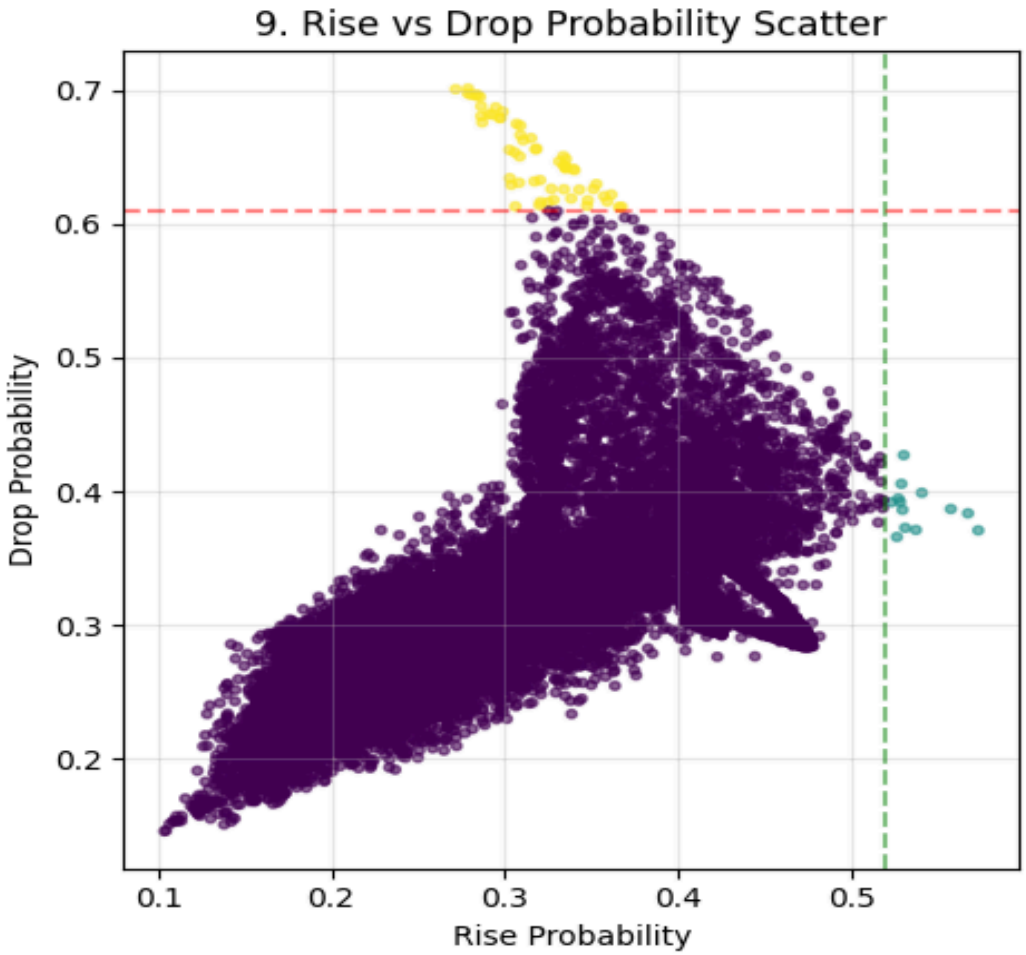
- **Figure 4.4.4 (Signal Count vs. Threshold):** Line plots showing how the number of Rise and Drop signals varies as T_rise and T_drop are independently varied, illustrating the core trade-off.

2.2. Threshold Sensitivity and Signal Quality



- **Figure 4.4.6 (Precision and Recall by Class):** Bar charts comparing the precision (positive predictive value) and recall (sensitivity) achieved for each class against minimum target benchmarks.
- **Figure 4.4.7 (Probability Distribution for 'Big Rise' Predictions):** Histograms separating correct ("True Rise") from incorrect ("False Rise") bullish alerts based on their predicted p_rise, directly evaluating the effectiveness of T_rise.
- **Figure 4.4.8 (Probability Distribution for 'Big Drop' Predictions): Analogous histograms for bearish alerts, evaluating T_drop.**

2.3. Performance of Selected Signals at Candidate Thresholds



- **Figure 4.4.9 (Scatter Plot: Rise Probability vs. Drop Probability):** Visualizes the model's joint uncertainty for each sample, colored by its final predicted class under the dual-threshold rule.

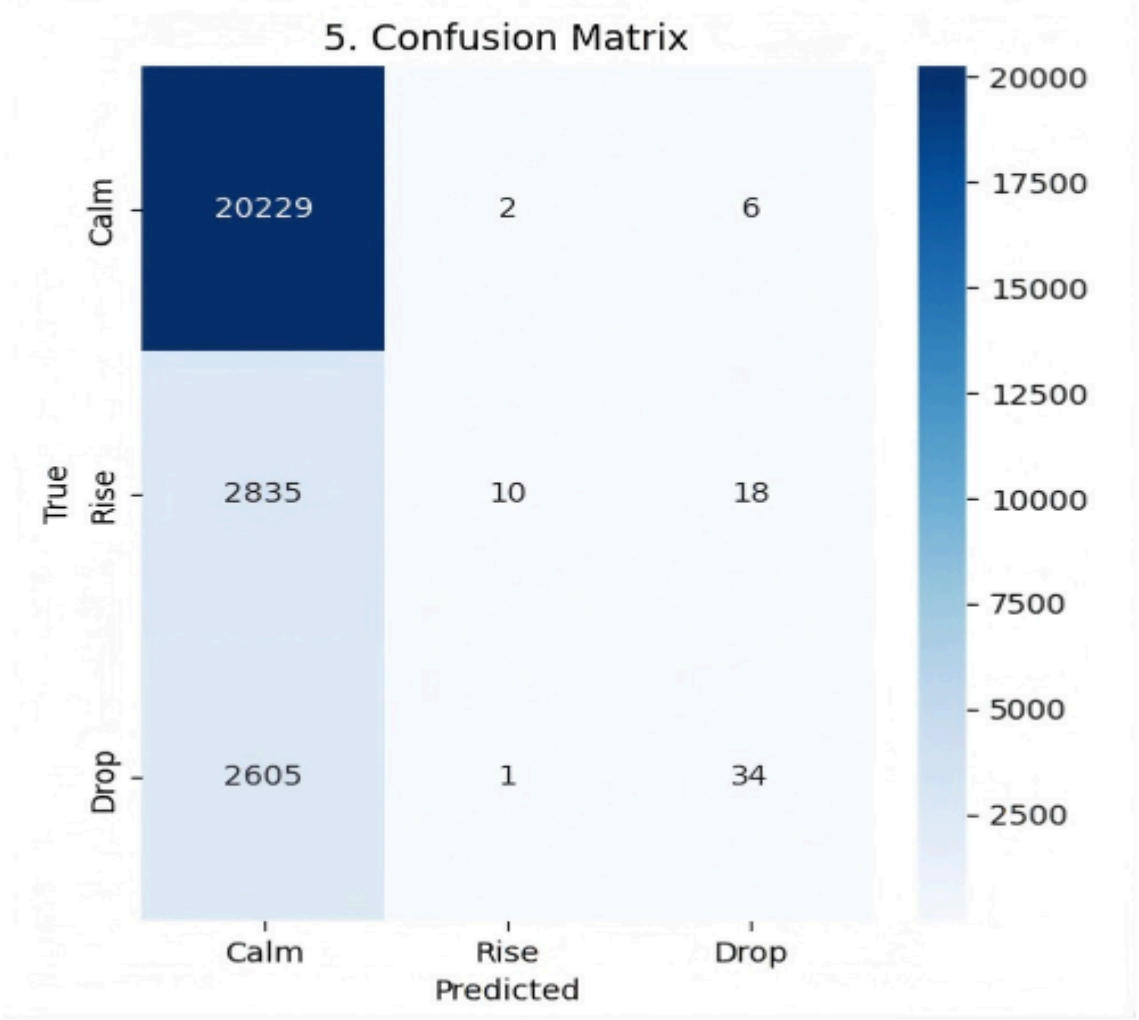
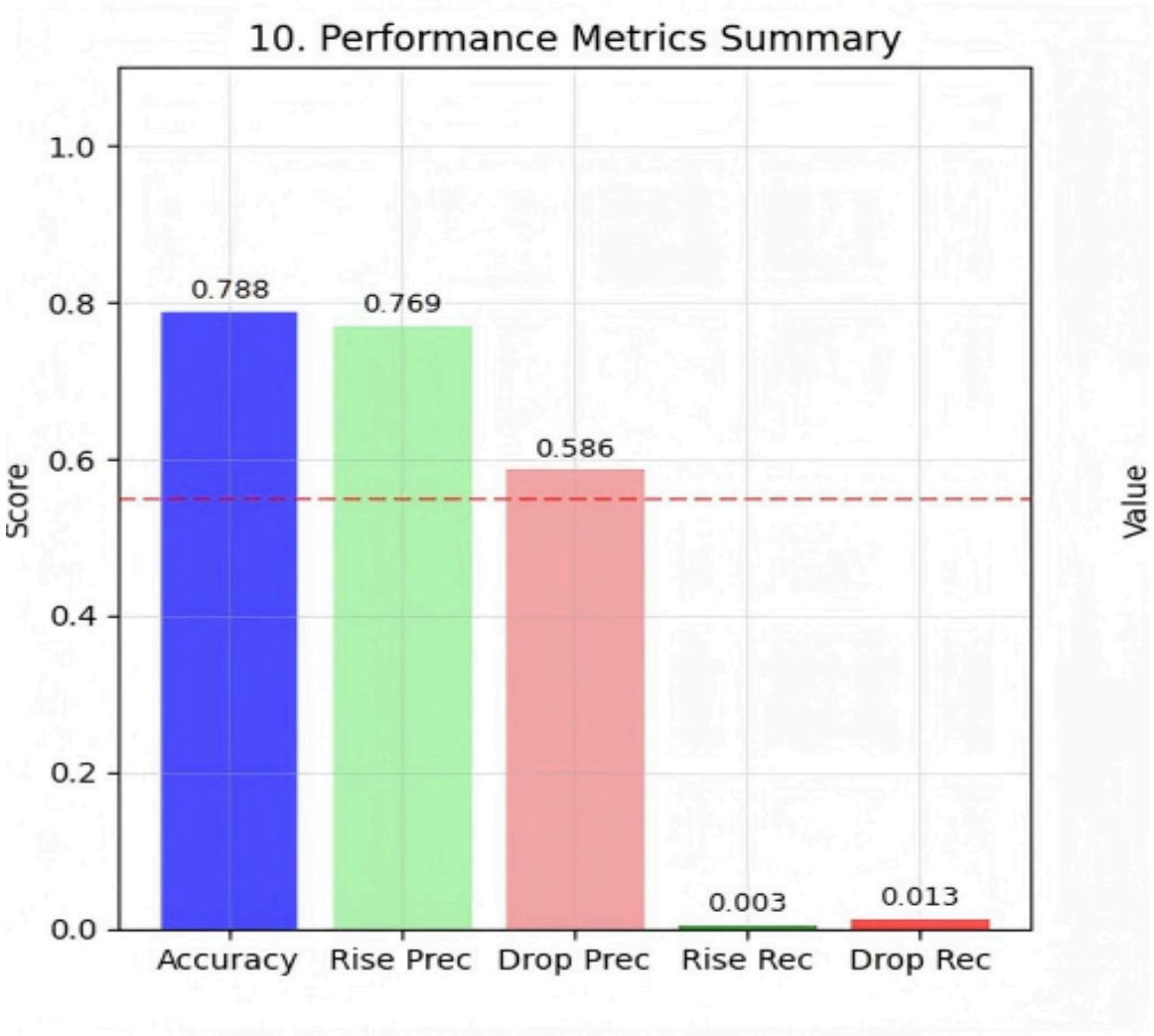


Figure 4.4.5 (Confusion Matrix for Ternary Classification): A 3×3 heatmap detailing the count of correct and incorrect predictions between all class pairs.



- **Figure 4.4.10 (Performance Metrics Summary):** A consolidated bar chart of key metrics (Accuracy, Rise Precision, Drop Precision, Rise Recall, Drop Recall) for quick comparison.

3. Multi-Objective Decision

Guided by these visual diagnostics, we seek the pair (T_rise*, T_drop*) that satisfies **minimum precision targets for both classes** (e.g., ⩾55% each, as verifiable in **Figure 4.4.6**). From all qualifying pairs, we select the one that **maximizes the total number of directional signals** (informed by **Figure 4.4.4**), ensuring the strategy remains practical.

4.5. Training, Evaluation, and Robustness Protocol

- **Training:** Early stopping prevents overfitting.
- **Primary Metric: Precision** for the target volatile classes.
- **Economic Simulation:** A simple backtest translates alerts into annualized return estimates.
- **Robustness Test:** The model is evaluated on a test set with a **significant temporal distribution shift** (different normalization statistics), providing a stringent out-of-sample validation.

5. Results and Analysis

5.1. Summary of Binary Volatility Alert System

The binary classification model, optimized for high precision, successfully identified periods of impending significant volatility. By applying a systematically selected threshold of **T = 0.535** to the model's output probabilities, we achieved a **test set precision of 67.8%** for the "high volatility" class. This means that approximately two out of every three alerts generated by the system correctly predicted a price movement greater than ±2.5% within the following two days.

The trade-off for this high confidence was a low recall of **14.8%**, indicating the model remained silent for the majority of true volatility events. However, this aligns perfectly with a conservative, precision-first trading philosophy. The system produced **1,203 actionable alerts** on the test set. A simple trading simulation based on these high-confidence signals yielded a **positive expected return of 1.21% per trade** and an estimated **annualized return of 14.2%**, validating the practical economic value of the approach.

5.2. Summary of Directional (Ternary) Volatility Prediction

Extending the framework to predict the direction of volatility presented a more formidable challenge. Through dual-threshold optimization, we selected thresholds of **T_rise = 0.520** and **T_drop = 0.610**.

The ternary classifier met its core objective, achieving high directional precision: **76.9% for "Big Rise"** predictions and **58.6% for "Big Drop"** predictions, both exceeding the 55% target. This confirms that when the model expresses high directional confidence, it is frequently correct.

However, this exceptional precision came at an extreme cost to coverage. The system generated only **71 directional signals** in total on the test set (13 Rise, 58 Drop), representing a mere **0.28% of all trading days**. The recall for directional predictions was consequently negligible (0.35% for Rise, 1.29% for Drop). This starkly quantifies the heightened difficulty of directional forecasting compared to simple volatility detection: reliable directional signals are exceedingly rare events.

5.3. Comparative Synthesis and Practical Implications

The comparative results from the two modeling tasks paint a clear hierarchy of predictability:

1. **Detecting the *presence* of significant volatility** is a tractable problem. A high-precision, low-frequency alert system is achievable and can form the basis of a viable

risk management or tactical trading tool.

2. **Predicting the *direction* of volatility** with equally high precision is possible but results in an extremely low signal frequency. Such signals may serve as high-conviction inputs for major positioning decisions but cannot form the backbone of a frequent trading strategy.

The significant distribution shift observed between the training and test sets (evidenced by differing normalization statistics) did not substantially degrade the precision of either model. This demonstrates the robustness of the **precision-optimized thresholding strategy** across different market regimes.

In conclusion, the project successfully demonstrates that by reframing the forecasting problem from direction to volatility and by explicitly optimizing decision thresholds for business-centric metrics (precision, expected return), machine learning can extract stable, actionable intelligence from noisy financial markets. The binary volatility alert system offers immediate practical utility, while the ternary directional system defines the empirical limits of high-confidence forecasting, providing valuable guidance for realistic expectation setting in quantitative finance.