StockPredictorAI: Quantitative Model & Technical Specification

Author: Omer Faruk Arar

Last Update Date: 21 October 2025

Abstract

This document lays out the core math and logic behind the StockPredictorAI platform. We use a unique **Proximity Scoring Model** that includes rules based on direction and time to create a reliable, measurable reputation score, the **Analyst Rating**. Our design focuses on data security, ensuring the scoring is fair, and preventing people from trying to exploit the reward system.

I. The Core Scoring Algorithm: Proximity Rating (R)

Our main way to measure accuracy is the **Proximity Rating (R)**. This score ranges from 0 to 100 and is calculated right when a prediction's deadline passes. This score is the basis for all rewards and reputation.

1. Directional Pre-Check (Binary Filter)

The first step checks if the predicted direction (Up or Down) was correct. If the analyst predicted the stock would rise, but it fell, the score is instantly zero.

$$D = \begin{cases} 0 & \text{if } (P_t - P_c) \cdot (A - P_c) < 0 \\ 1 & \text{otherwise} \end{cases}$$

Where:

- D: Directional Accuracy (0 or 1).
- P_t: Predicted Target Price.
- P_c: Price at Creation (The price when the prediction was made).
- A: Actual Price at Deadline (The price used for the final check).

If $D = 0$, the rating R is immediately 0.

2. Proximity Score Calculation (Normalization)

If the direction was correct ($D = 1$), the **raw score** R_{raw} is calculated based on how far the actual price (A) was from the predicted price (P_t), measured against our maximum accepted error, ϵ.

$$\text{Error Percentage} = E_p = \frac{|P_t - A|}{A}$$

The raw rating R_{raw} is defined as:

$$R_{\text{raw}} = \begin{cases} 100 \cdot \left(1 - \frac{E_p}{\epsilon}\right) & \text{if } E_p \leq \epsilon \\ 0 & \text{if } E_p > \epsilon \end{cases}$$

Constraint: We set the maximum tolerable error ϵ at 20 (0.20).

3. Time Penalty Optimization (Capping)

To prevent specific market abuse patterns (e.g., 'sniping' a prediction minutes before close), we apply a **Time Penalty Function** $\mathcal{P}(t)$. This reduces the maximum attainable score (R_{max}) as the prediction window elapses.

The final rating cap is defined as:

$$R_{\text{max}} = 100 - \mathcal{P}(t, k)$$

Where k represents the **Prediction Type** (e.g., Hourly, Daily).

The Penalty Function $\mathcal{P}(t, k)$

The penalty is calculated based on the elapsed time t_e since the start of the prediction window, subject to a **Grace Period (T_g)** and a **Maximum Penalty Cap (L_{max})**.

$$\mathcal{P}(t_e) = \begin{cases} 0 & \text{if } t_e \leq T_g \\ L_{\text{max}} \cdot \left(\frac{t_e - T_g}{T_{\text{total}} - T_g}\right) & \text{if } t_e > T_g \end{cases}$$

System Constants:

Prediction Type (k)	Grace Period (T_g)	Max Penalty ($L_{\max}$)	Implication
Hourly	**10 Minutes**	**20 Points**	First 10 mins are risk-free. Late entry max score = 80.
Daily	**0 Minutes**	**20 Points**	Linear decay (segmented in 20 steps).
Weekly	**0 Minutes**	**20 Points**	Linear decay from hour zero.
Monthly	**0 Minutes**	**25 Points**	Max penalty scales to 25.
Quarterly	**0 Minutes**	**25 Points**	Max penalty scales to 25.
Yearly	**0 Minutes**	**30 Points**	Strict long-term conviction required. Max penalty scalar = 30.

Note: For **Daily** predictions, T_{total} is dynamically segmented into 20 market-hour intervals to ensure fairness across different timezones.

Case Study: Intra-Day Decay Analysis

Consider a scenario under standard market conditions (Opening time $T_{open} = 09:30$).

- **Parameters:** Prediction Type is **Daily** ($T_{\text{total}} \approx 390$ min, $L_{\max} = 20$).
- **Action:** Analyst submits prediction at $T_{sub} = 10:00$.
- **Elapsed Time (t_e):** $10:00 - 09:30 = 30$ min.

Substituting into $\mathcal{P}(t)$:

$$\mathcal{P}(30) = 20 \cdot \left(\frac{30}{390}\right) \approx 1.54$$

Applying the floor function for integer scoring:

$$R_{\max} = 100 - \lfloor 1.54 \rfloor = 99$$

Conclusion: The analyst retains 99 of potential reputation points, reflecting high conviction despite the 30-minute delay.

II. Algorithmic Integrity & Misuse Prevention

The system is designed to be fair and auditable, specifically by adding risk-based weighting to prevent abuse of the reward structure.

1. Weighted Target Hit Bonus

We reward analysts for hitting their target price mid-period (Target Hit Validation). To ensure this doesn't lead to spamming short-term predictions, the reward is dynamically scaled based on the prediction's duration.

$$\text{Bonus Points} = B \cdot W_{\text{type}}$$

Where:

- B: Base Target Hit Bonus (e.g., 5 points).
- W_{type}: Prediction Type Weight (a scaling factor).

Prediction Type	Risk/Effort Factor	Weight (W_{type})	Purpose
Hourly	Low Volatility, High Frequency	0.5	Mitigation: Reduces volume abuse.
Daily	Standard	1.0	Baseline reward.
Weekly	Medium Term	2.0	Reward for multi-day foresight.
Monthly	High Volatility	4.0	Significant reward for 30-day accuracy.
Quarterly	Strategic	6.0	Major quarterly forecast reward.
Yearly	High Risk, Macro Forecasting	10.0	Incentivization: Rewards long-term conviction significantly.

2. Database Transactionality (Atomicity)

If an assessed prediction needs to be removed by an administrator, the platform must correctly and safely subtract the points previously awarded. This is handled using MongoDB's advanced transaction features, ensuring complete ACID compliance (all or nothing).

Rollback Algorithm (Key Steps): When an Admin deletes a prediction:

1. **START TRANSACTION**
2. Read the points awarded (P_{awarded}) from the Prediction record.
3. Subtract Points: $\text{User.AnalystRating.Total} \leftarrow \text{User.AnalystRating.Total} - P_{\text{awarded}}$.
4. Delete Prediction Record.
5. **COMMIT TRANSACTION**

This sequence guarantees that the user's reputation index stays accurate even when data is manually removed.