---

## 📘 **Final Price Prediction Model – Mathematical Breakdown**

---

### 1. Notation

Let:

- $P_t$: Actual closing price at historical time $t$
- $F_t = [P_t, EMA_{20,t}, MACD_t, RSI_{MA_t}, RSI_{Slope_t}, OBV_t, ATR_t, StdDev_t, BBWidth_t]$:

  Raw feature vector at time $t$
- $Scaler(F_t) = f_t$: MinMax-scaled feature vector
- $Scaler_{close}^{-1}(p')$: Inverse transform for scaled closing price
- $LSTM(X; \Theta)$: LSTM function with learned parameters Θ, returning next scaled price
- $k$: Lookback window size
- $T$: Last time index of historical data
- $N$: Number of future time steps to predict
- $Date_{T+i}$: Calendar date for the $i-th$ forecast
- $S_d$: Historical sentiment score for day $d$
- $\hat{S}_{future,d} = ForecastSentiment(d)$: Predicted sentiment using linear regression
- $w_s$: Sentiment weight factor
- $P_{pred,i}$: Final predicted price for day $T + i$

---

### 2. Stage-by-Stage Prediction Process

---

### Stage 1: LSTM Core Prediction

- Initial Input Sequence:

$$X_0 = [f_{T-k+1}, f_{T-k+2}, \dots, f_T]$$

- First Scaled Prediction:

$$\hat{y}'_1 = LSTM(X_0; \Theta)$$

For $i = 2$ to $N$:

- Feature Estimation:

$$f'_{i-1} = ConstructFeatures(\hat{y}'_{i-1}, X_{i-2})$$

- Updated Input Sequence:

$$X_{i-1} = [f_{T-k+i}, \dots, f_T, f'_1, \dots, f'_{i-1}]$$

- Scaled Prediction:

$$\hat{y}'_i = LSTM(X_{i-1}; \Theta)$$

---

**Stage 2: Inverse Scaling**

Convert scaled predictions to original price space:

$$\hat{y}_i = Scaler^{-1}_{close}(\hat{y}'_1)$$

---

**Stage 3: Initial Price Alignment**

To prevent a visual jump:

- Compute alignment shift:

$$Shift = P_T - \hat{y}_1$$

- Adjusted prices:

$$AdjustedPrice_i = \hat{y}_i + Shift$$

---

**Stage 4: Sentiment Adjustment**

If sentiment analysis is enabled:

- Forecast sentiment:

$$\hat{S}_{future,i} = ForecastSentiment(Date_{T+i})$$

- Sentiment multiplier:

$$SentimentFactor_i = 1 + w_s \cdot \hat{S}_{future,i}$$

- Adjusted prices:

$$SentimentAdjustedPrice_i = AdjustedPrice_i \cdot SentimentFactor_i$$

Otherwise:

$$SentimentAdjustedPrice_i = AdjustedPrice_i$$

---

**Stage 5: Noise Injection**

Simulates market volatility:

- Mean price:

$$\mu_{pred} = \frac{1}{N}\sum_{i=1}^{N} SentimentAdjustedPrice_i$$

- Noise standard deviation:

$$\sigma_{noise} = \mu_{pred} \cdot volatility_{factor}$$

- Add noise:

$$Noise_i \sim N(0, \sigma_{noise}^2)$$

$$NoisyPrice_i = SentimentAdjustedPrice_i + Noise_i$$

---

**Stage 6: Price Floor Enforcement**

Ensure prices remain realistic:

$$P_{pred,i} = max\,(NoisyPrice_i, 0.01)$$

---

**3. Final Unified Mathematical Expression**

$$P_{pred,i} = max\,((Scaler_{close}^{-1}(LSTM(X_{i-1}; \Theta)) + (P_T - Scaler_{close}^{-1}(LSTM(X_0; \Theta)))) \cdot (1 + w_s \cdot \hat{S}_{future,i}) + Noise_i,\ 0.01)$$

---

### 4. Analysis and Insights

✅ **Strengths**

- Combines deep learning (LSTM) with technical indicators and sentiment.
- Sentiment forecasting is modular and interpretable.
- Visual continuity and stochastic realism handled by alignment + noise.
- Scalable to multiple assets or extended timeframes.

⚠️ **Limitations**

- Feature re-use post-step 1 may compound drift.
- Shift alignment introduces a manual correction bias.
- Sentiment impact is modeled linearly, which may oversimplify market psychology.
- Fixed noise variance doesn't account for changing market volatility.

---