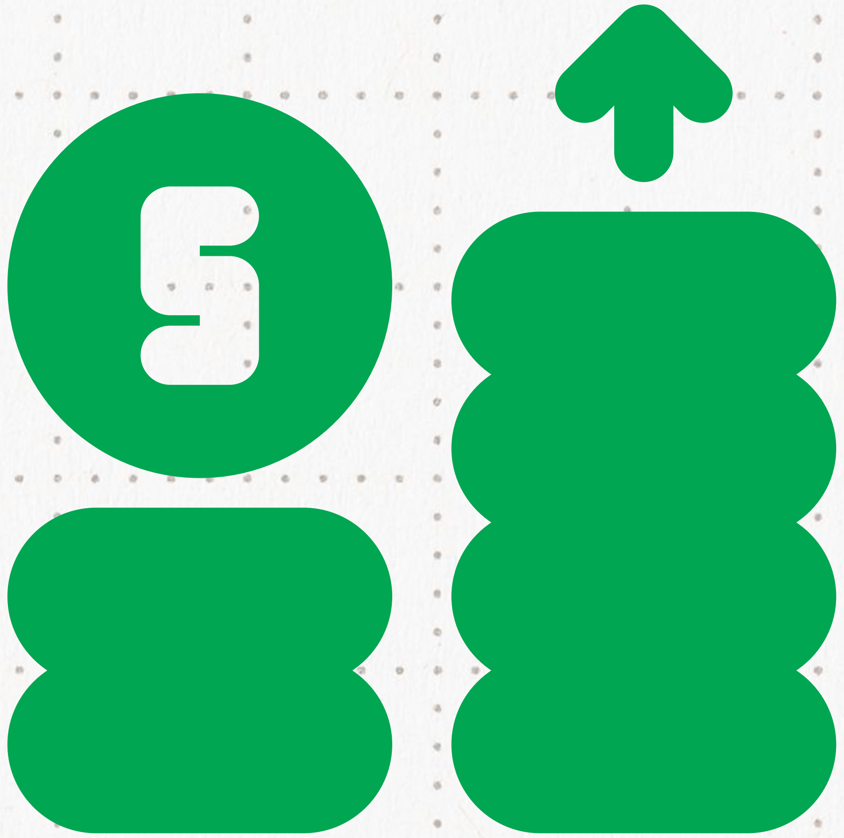
AI-Enhanced Stock Forecasting Decision Support System

Group 1A

1093532 石聰
1103544 陳錦顥
1103558 林米克
1103559 諾吉
1103553 奧馬爾

What we'll learn today

Introduction



Challenges in Traditional Financial Decision-Making

Traditional methods and BI tools struggle with biases, rigidity, and lack of adaptability, leading to errors in dynamic markets and significant financial losses for investors.

Our LSTM-Based Solution

Our LSTM model leverages real-time data to enhance precision, empowering investors with personalized, risk-aligned insights for adaptive and effective financial decision-making.

Existing Solutions

Stock market prediction always has been a developing research area

Current state

Developing research area

Machine Learning Integration

Division

Existing models

Time Series Models

LSTM (Long Short-Term Memory)

GRU (Gated Recurrent)

Multi-Modal Models

CNN-LSTM hybrids

Models combining price data with BERT

Ensemble Approaches

Random Forests and XGBoost

Multiple neural networks

Challenges

Market Adaptability

Markets are adaptive systems - if a predictable pattern emerges, traders quickly exploit it until it disappears.
The very act of using a prediction tool can change market behavior, invalidating its predictions

Data Complexity

Stock prices are influenced by countless factors: global events, company performance, human psychology, regulations, natural disasters, etc.

Signal vs Noise Problem

Market data contains a huge amount of random noise. Models often end up "learning" the noise instead of meaningful patterns

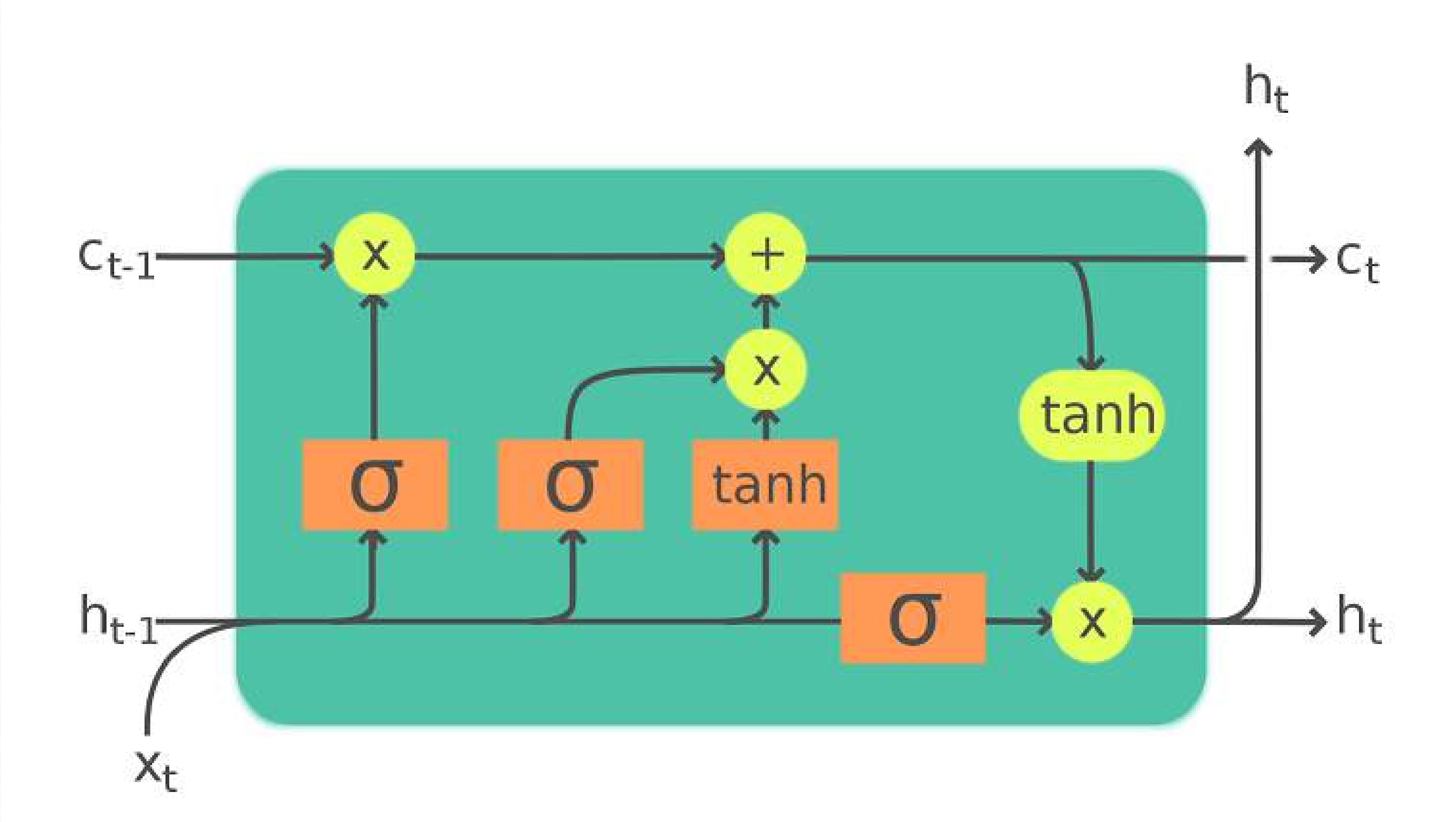
Framework

1. Data Preprocessing
2. Data Splitting
3. Model Architecture
4. Model Training
5. Evaluation

FOUNDATION

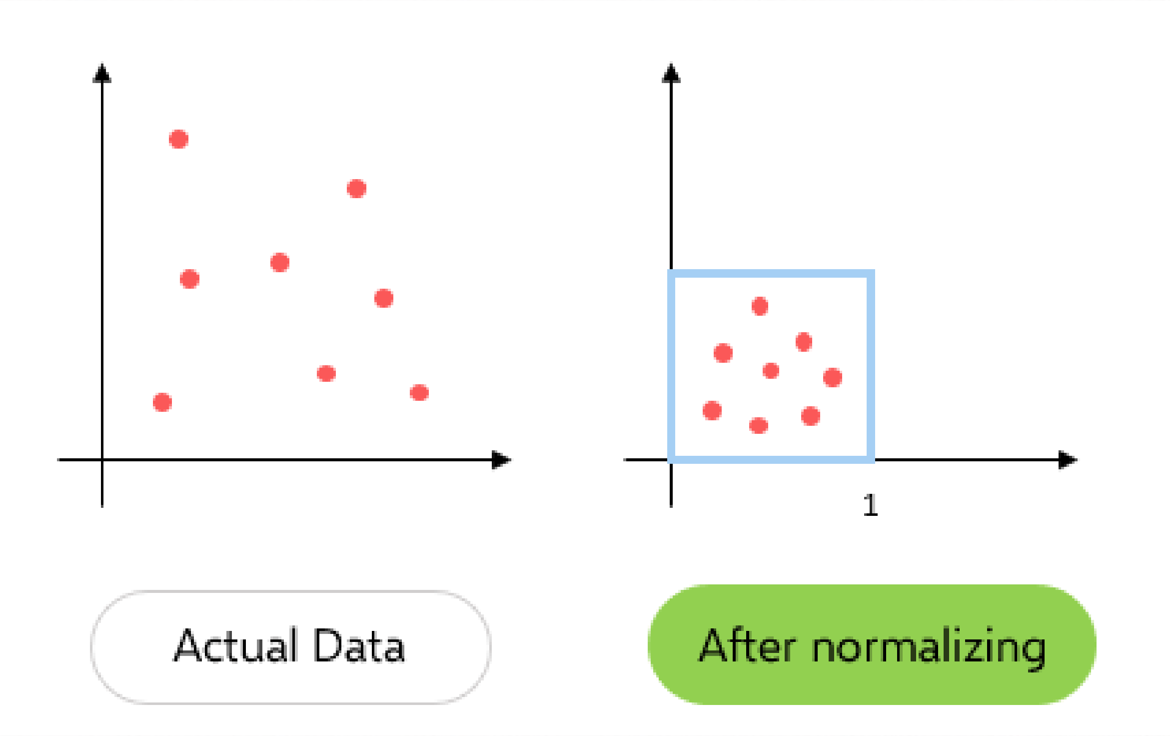
LSTM NEURAL NETWORKS TAILORED FOR TIME-SERIES PREDICTION

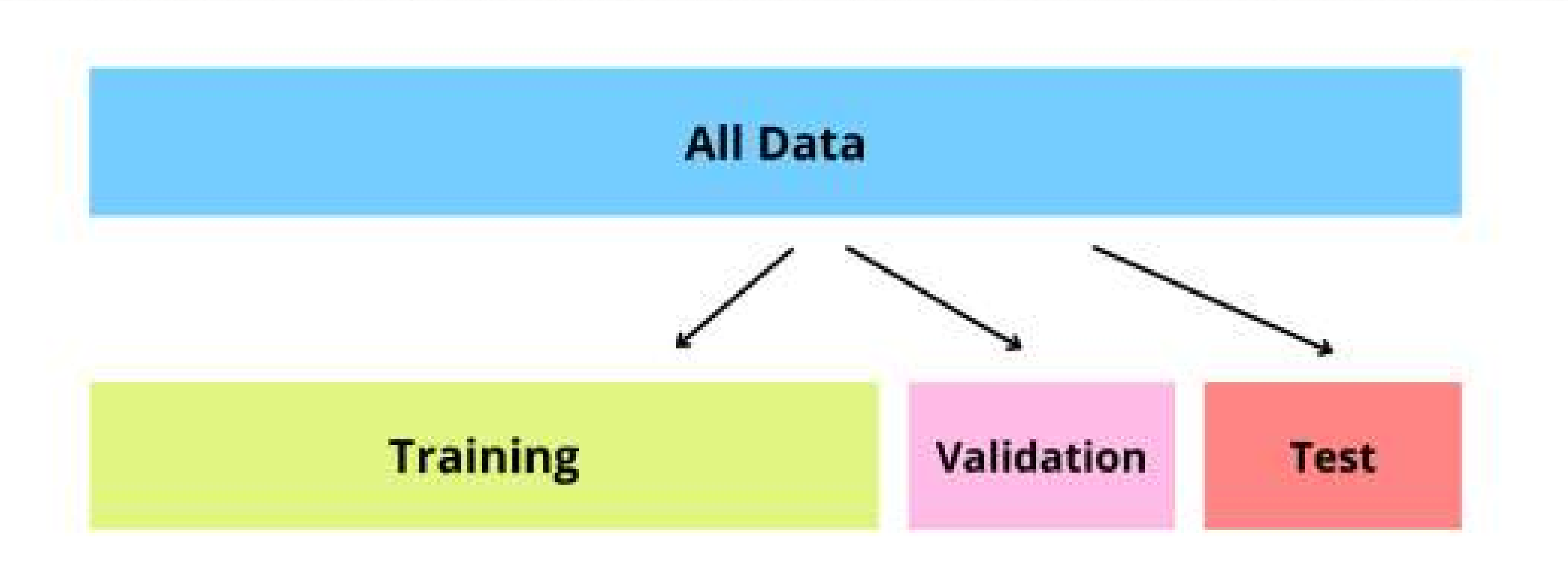
- Uses a deep learning framework involving Long Short-Term Memory (LSTM) neural networks to predict stock market prices.

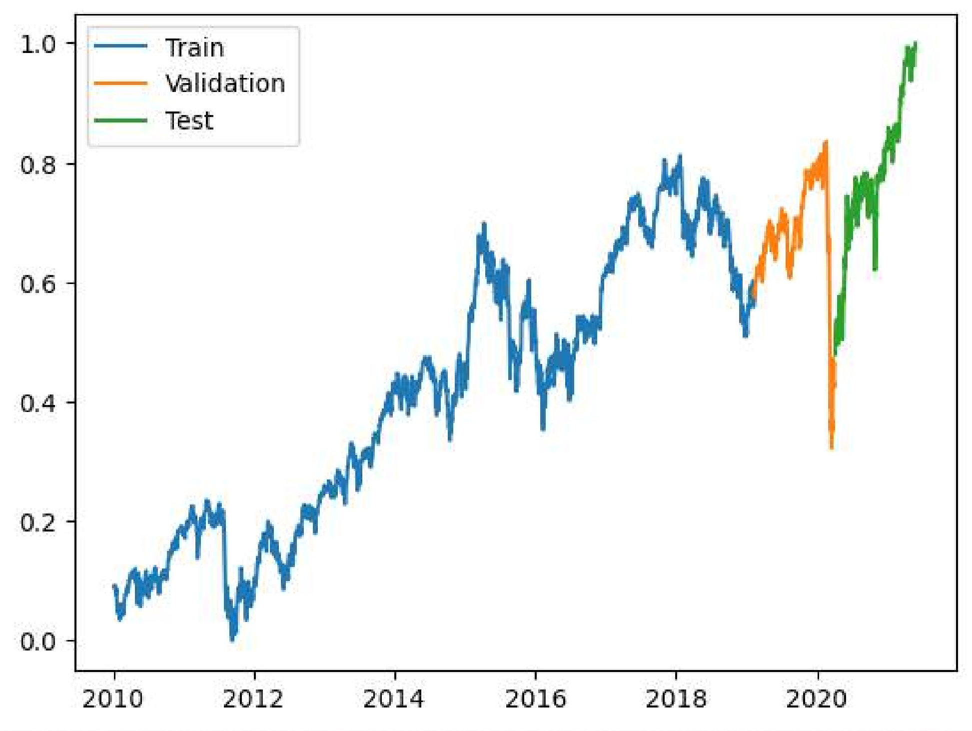


DATA HANDLING

Preprocessing to normalize and structure data.

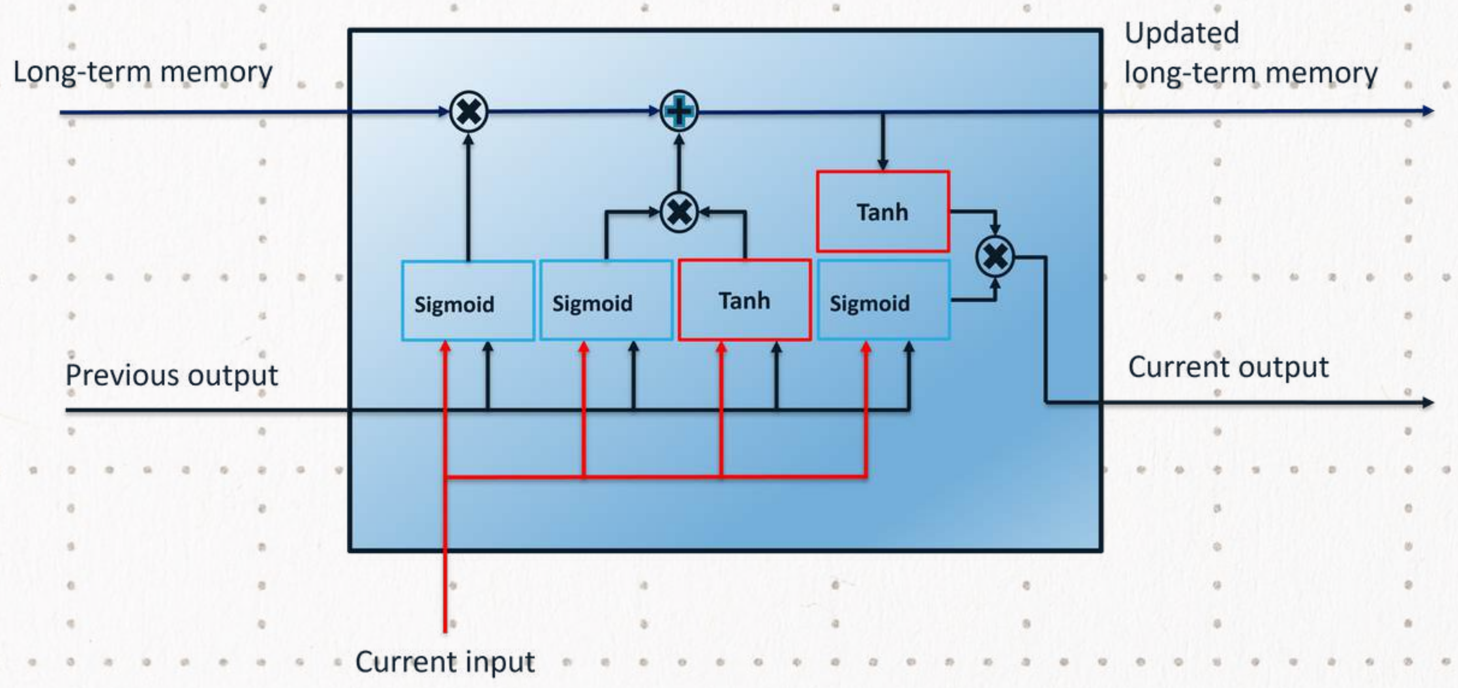




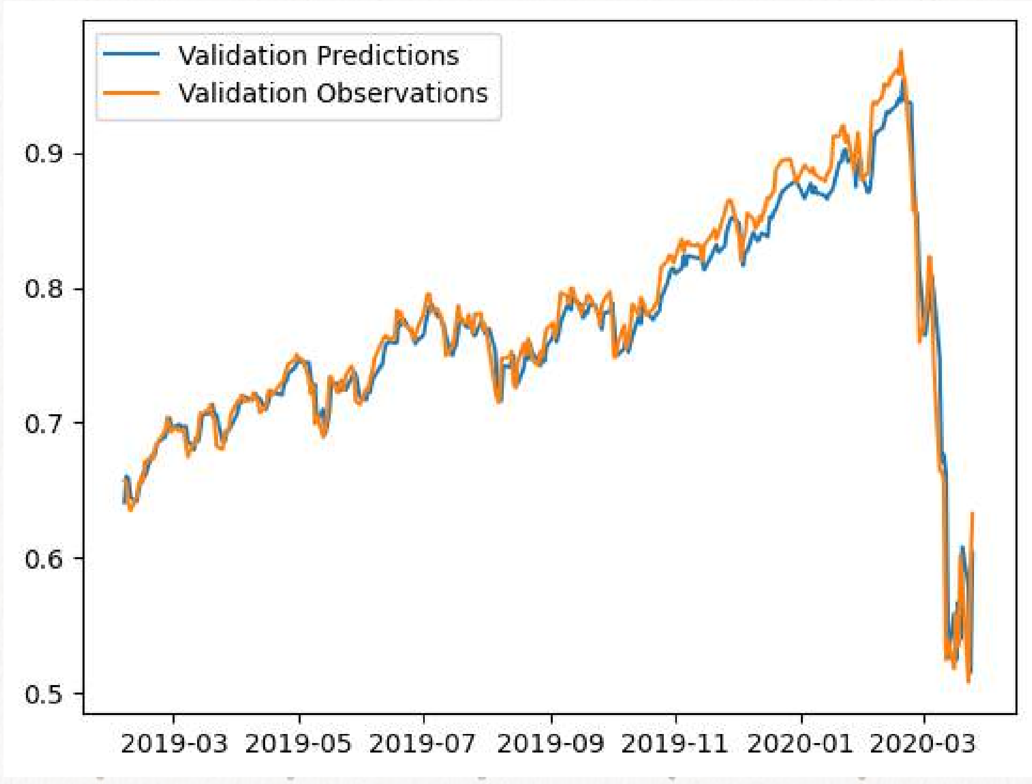


CORE MODEL ARCHITECTURE

Sequential LSTM layers to capture stock price trends.



Output layers predicting future stock prices.



IMPLEMENTATION TOOLS

Python Libraries:
Tensorflow/Keras, Pandas,
Numpy

Environment: Google Colab

EVALUTATION METRICS

MODEL LOSS VALUES ON TEST DATASETS.

Mean Squared Error (MSE)

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

where yi is the actual value, y^i is the predicted value, and n is the number of predictions.

PERFORMANCE FOR INDIVIDUAL STOCKS.

- Post-training, the model is tested on unseen data for each stock index.
- The framework outputs the loss values to determine the model's performance

Methodolgy

Data Collection

Handling missing values, outliers, and inconsistencies.

Scaling numerical features and transforming time-series data for training

Model Development

Perform cross-validation and hyperparameter tuning to refine models.

Select the best-performing model(s) for deployment.

Integration and Deployment

Integrate the Tkinter UI with the model to form the system.

Host locally for efficient and controlled performance.

Future Improvement

Integrate real-time stock data for timely decision-making.

Explore cloud deployment for broader accessibility.

Possible Benefits

Accurate Predictions

- Multiple features and big data increases prediction accuracy
- Economic indicators, and social and poltical factors increase reliability of predictions

Reduced Bias

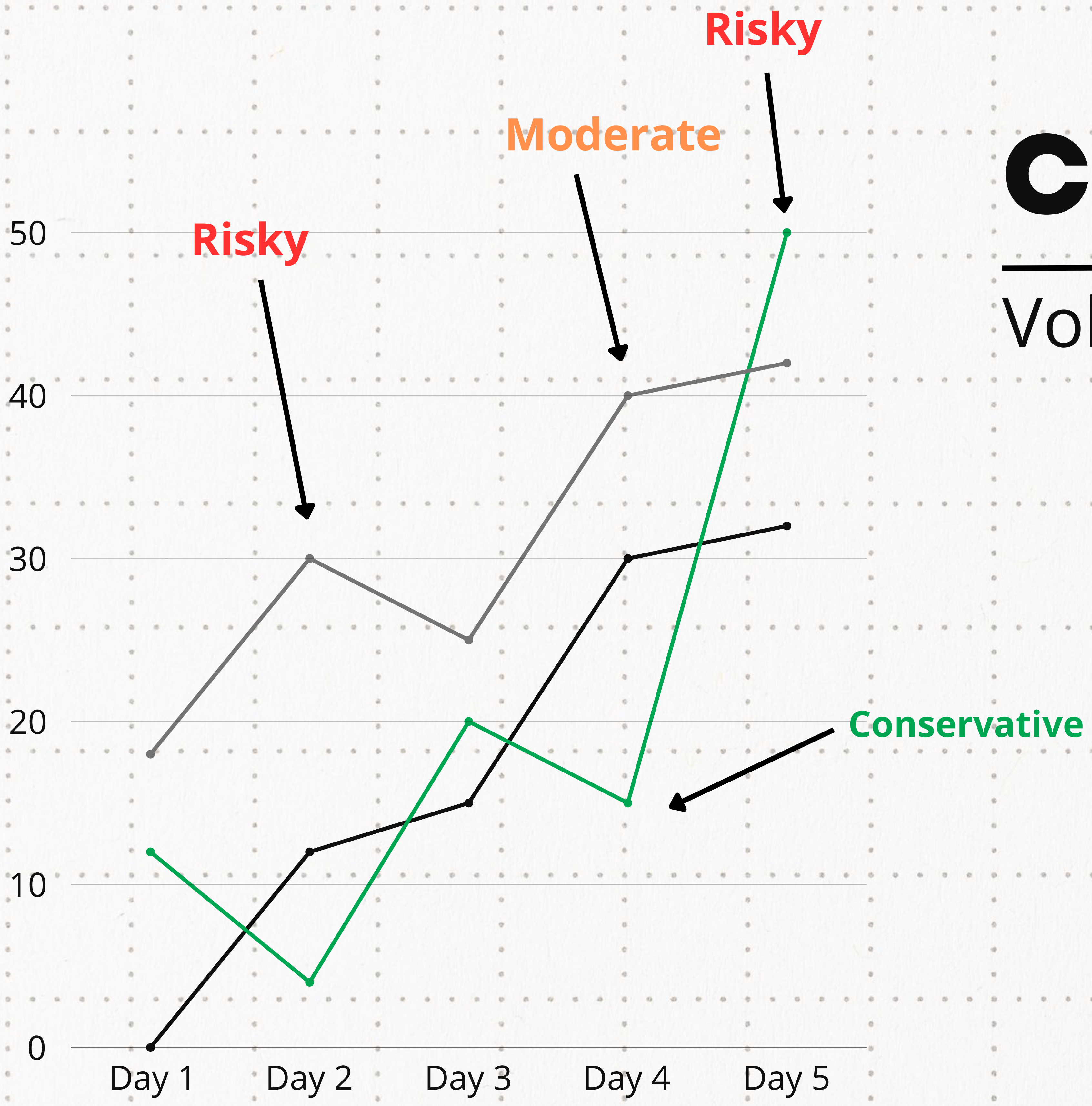
- AI informs the user of their risks
- Helps to remove emotion and previous bias from decision making

Decreased Risk

- Big data allows for consideration of all possibilites
- Can be adjusted to fit the risk tolerance of the user

Stock Classification

Stock Classification serves as a useful insight for traders.



Classifying Stock

Volatility-Based Classification

Volatility refers to the rate and magnitude of price movements for a stock.

It measures how unpredictable or unstable a stock's price is compared to its historical average.

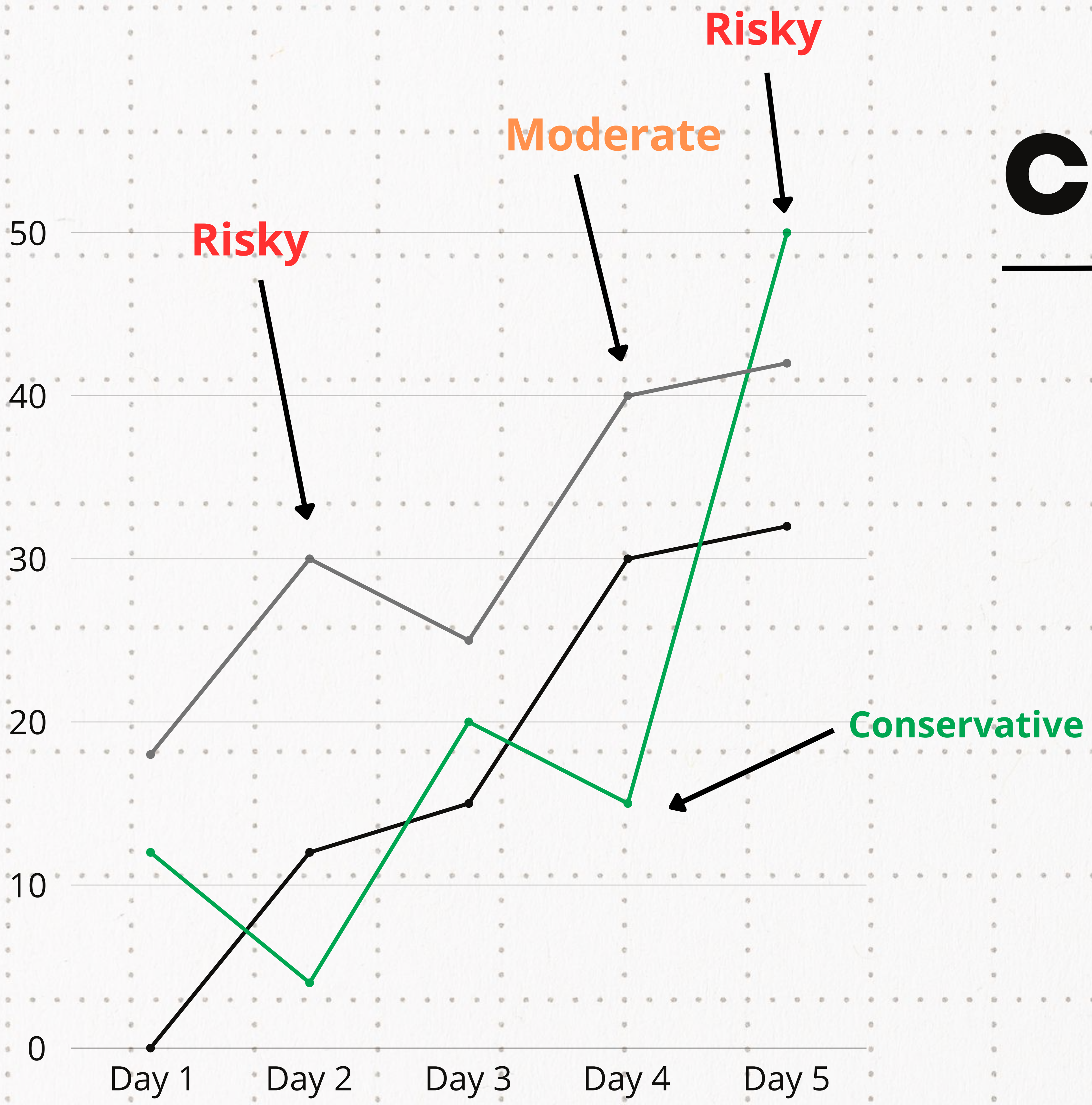
Why Volatility Matters:

Risk Indicator:

- High volatility often implies greater investment risk
- Low volatility indicates stability, making the stock potentially less risky.

Potential for Profit:

- High volatility can provide opportunities for traders to profit from rapid price swings.



Classifying Stock

Thresholds for Classification:

Risky: Volatility > 10%

Moderate: 5% ≤ Volatility ≤ 10%

Conservative: Volatility < 5%.

$$\text{Volatility (\%)} = \left(\frac{\text{Standard Deviation}}{\text{Mean}}\right) \times 100$$

Given an example of predicted prices:

If window size = 3

$$[100, 102, 104, 103, 101, 99, 97]$$

$$\text{Mean} = \frac{100+102+104}{3} = \frac{306}{3} = 102$$

$$\text{Std Dev} = \sqrt{\frac{(100-102)^2+(102-102)^2+(104-102)^2}{3}} = \sqrt{\frac{4+0+4}{3}} = \sqrt{\frac{8}{3}} \approx 1.63$$

$$\text{Volatility} = \left(\frac{1.63}{102}\right) \times 100 \approx 1.60\%$$

Key Observations:

Predicted Prices from a test dataset:

Classifying on normalized prices:

- Classification throughout the dataset varies more.
- Normalized data emphasizes small price fluctuations, resulting in more varied classifications.

Classifying on denormalized prices:

- Classification throughout the dataset varies less.
- Denormalized data minimizes small fluctuations relative to the larger scale, making classifications more uniform.

Classification on normalized prices:

Predicted Price	Mean	Std. Deviation	Classification
0.724341	0.76356	0.025271	Conservative
0.723366	0.755295	0.02786	Conservative
0.746781	0.750208	0.025215	Conservative
0.772221	0.748787	0.023322	Conservative
0.80961	0.753341	0.03157	Conservative
0.815472	0.760371	0.039428	Moderate
0.823106	0.773557	0.043151	Moderate
0.834979	0.789362	0.042375	Moderate
0.828033	0.804315	0.032528	Conservative
0.851011	0.819205	0.024722	Conservative
0.844778	0.82957	0.015064	Conservative

Classification on denormalized prices:

Predicted Price	Mean	Std. Deviation	Classification
9597.379	9857.56	167.6515	Conservative
9590.906	9802.729	184.8298	Conservative
9746.25	9768.986	167.2794	Conservative
9915.024	9759.557	154.7229	Conservative
10163.07	9789.767	209.4394	Conservative
10201.96	9836.408	261.5712	Conservative
10252.61	9923.884	286.27	Conservative
10331.37	10028.74	281.1224	Conservative
10285.29	10127.94	215.7996	Conservative
10437.73	10226.72	164.0106	Conservative
10396.38	10295.49	99.93789	Conservative

Key Takeaways:

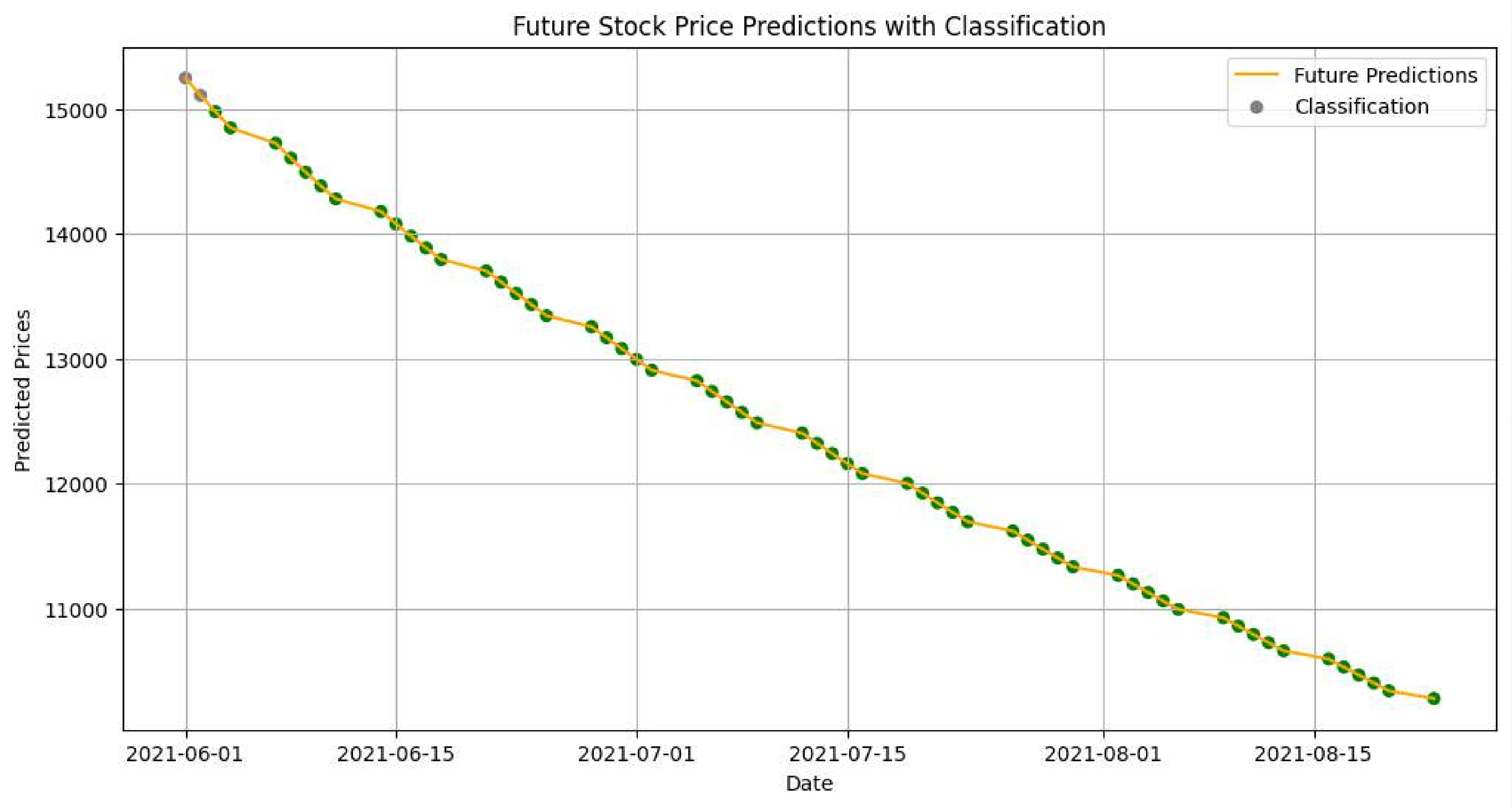
Classification on Normalized Prices:

- Apply classification before denormalizing the predicted prices for more realistic results.

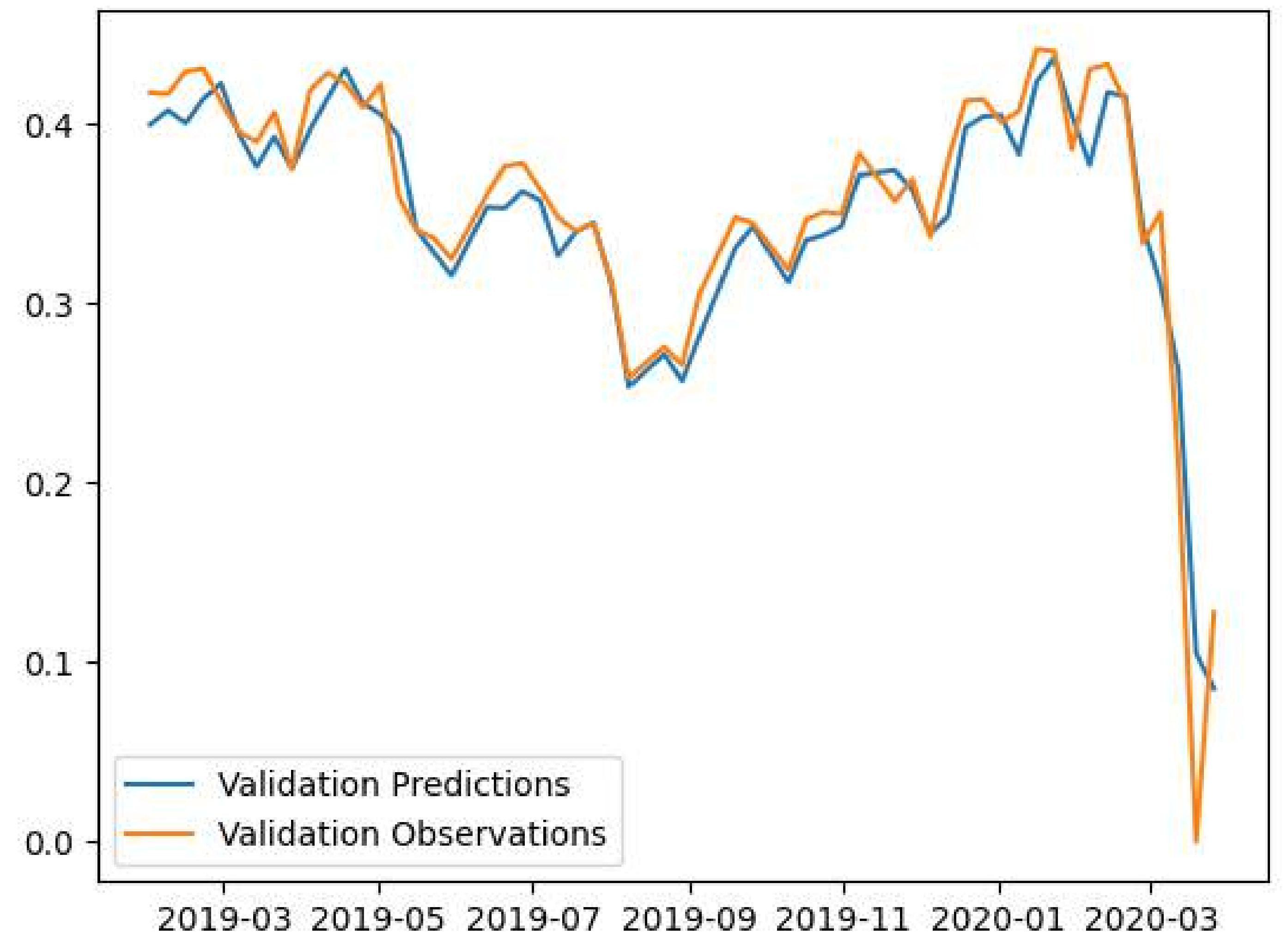
Volatility Sensitivity:

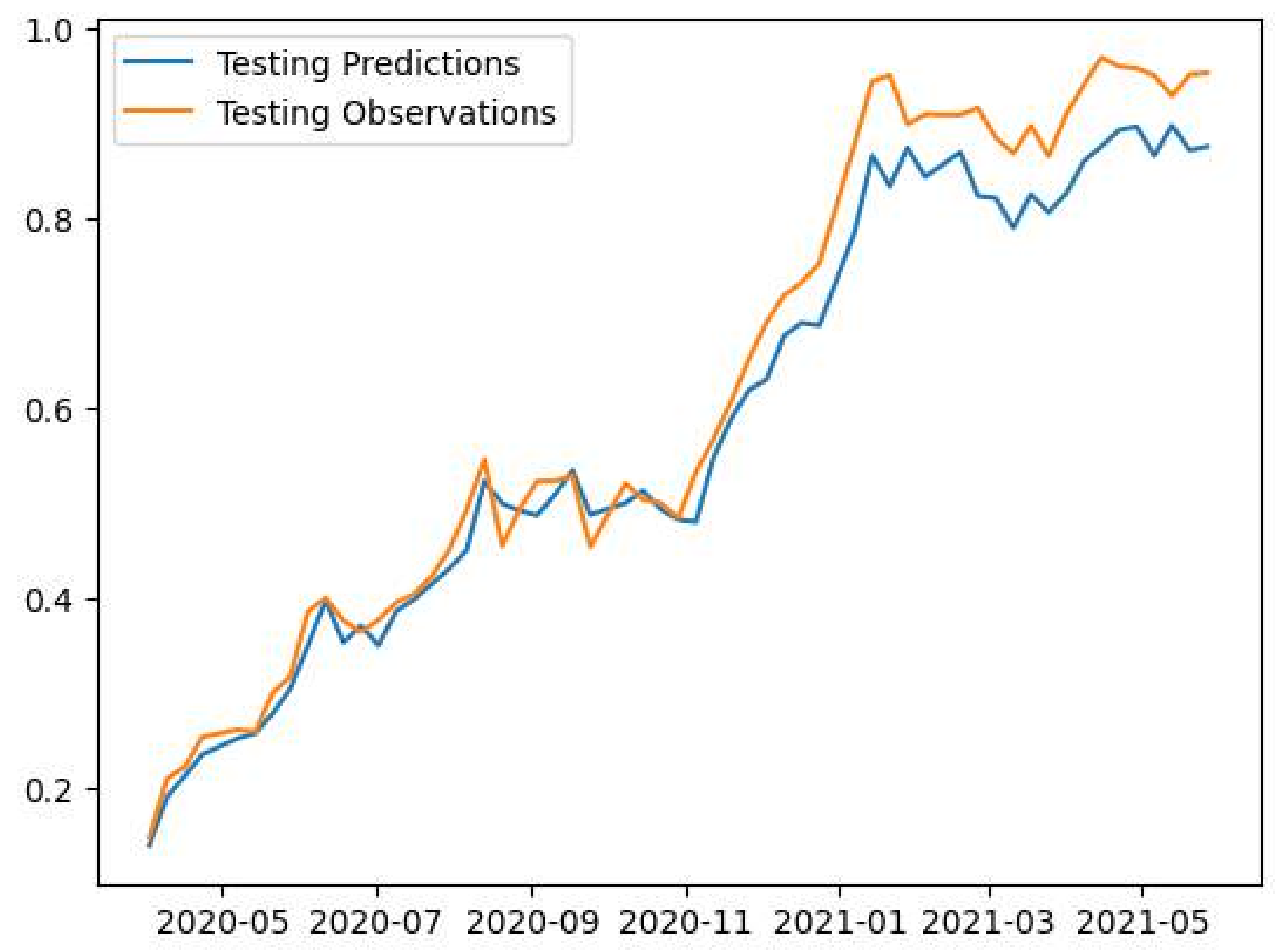
- **Challenge**: The rolling window size and thresholds for volatility classification significantly impact the results.
- **Actionable Insight**: Experiment with different window sizes and adjust thresholds to align with the scale and behavior of the data.

Predicted Prices 60 days into the future:



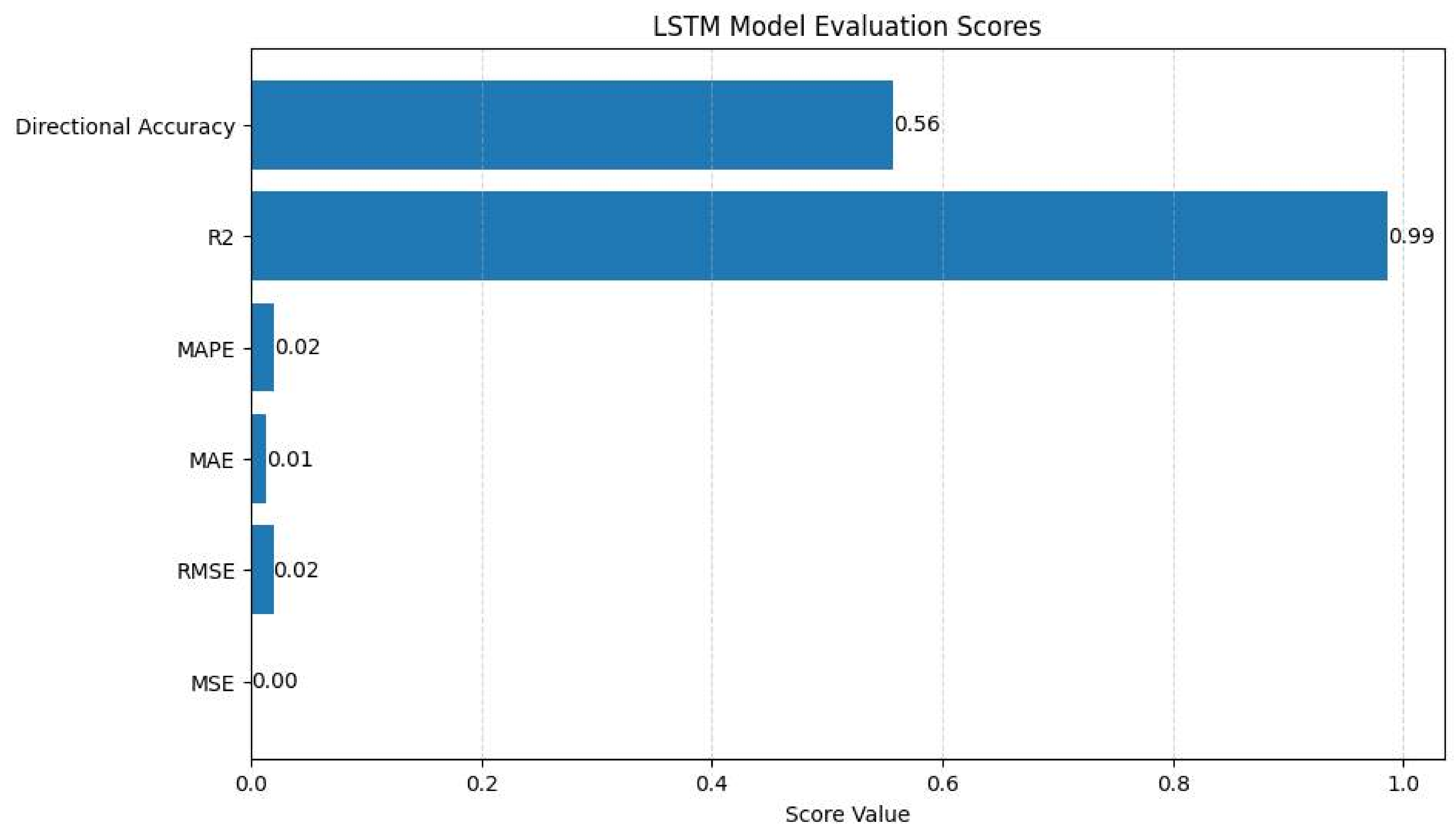
Results





- **Validation and Testing predictions are close to observations**
- **Predictions show a slight lag behind observations**
- **While the predictions are close, it may be too late to buy or sale by the time they're made**

Metrics



- **Directional accuracy is a coin flip**
- **R2 score handles variations well**
- **Remaining scores are better than average**

Conclusion

- Datasets need updating
  - Older datasets will impact predictions
- Generalization needs improving
  - Predicting actual values vs rate of change
- Limted Features
  - Only used closing price
  - Other socioeconomic and political factors should be considered
- Classification algorithm
  - When to classify?
    - Before or after denormalization

Demo

Thank You