Can a Neural Network Find Patterns in Market Chaos?

An End-to-End Case Study on Forecasting TSMC Stock Trends with an LSTM Model.

TSMC (2330.TW)

NotebookLM

The Goal: Predict Tomorrow's Price *Direction*, Not the Exact Price

Financial markets are notoriously difficult to predict. Our objective is not to forecast a specific closing price, but to solve a more focused binary classification problem: Will TSMC's stock close 'Up' or 'Down' on the next trading day?

Based on historical data, can we build a model with predictive power greater than a 50/50 random chance?

NotebookLM

Our Four-Step Process for Building and Evaluating the Model

1. Prepare the Data

Sourcing, cleaning, and structuring the raw 2024 TSMC trading data.

2. Design the Network

Architecting a Long Short-Term Memory (LSTM) network to recognize temporal patterns.

3. Train & Evaluate

Training the model on 80% of the data and testing its performance on the remaining 20%.

4. Uncover the Insight

Analyzing the results beyond headline metrics to understand the model's true behavior.

The Foundation is One Year of TSMC's Daily Trading Data

- **Source:** `TSMC_2330_2024.csv`
- **Timeframe:** January 2, 2024 – late November 2024
- **Raw Features:** Open, High, Low, Close, Volume (OHLCV)

Date	Open	High	Low	Close	Volume
2024-01-02	586.00	593.00	585.00	593.00	34,506,000
2024-01-03	589.00	589.00	578.00	579.00	45,621,000
2024-01-04	577.00	581.00	575.00	580.00	29,864,000
2024-01-05	578.00	580.00	575.00	576.00	27,998,000
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

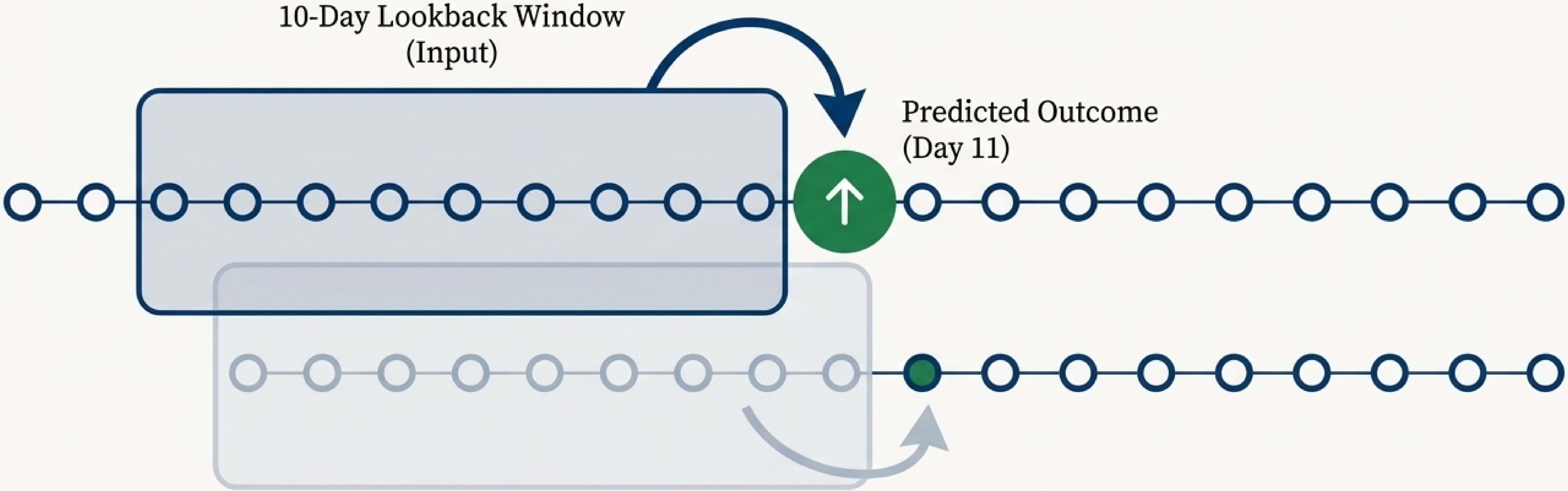
We Transformed Raw Data into a Learnable Format

1. **Define the Target**: We engineered a binary `Target` variable. A value of ‘1’ means the next day’s close was higher than the current day’s close (`Close_t+1 > Close_t`).

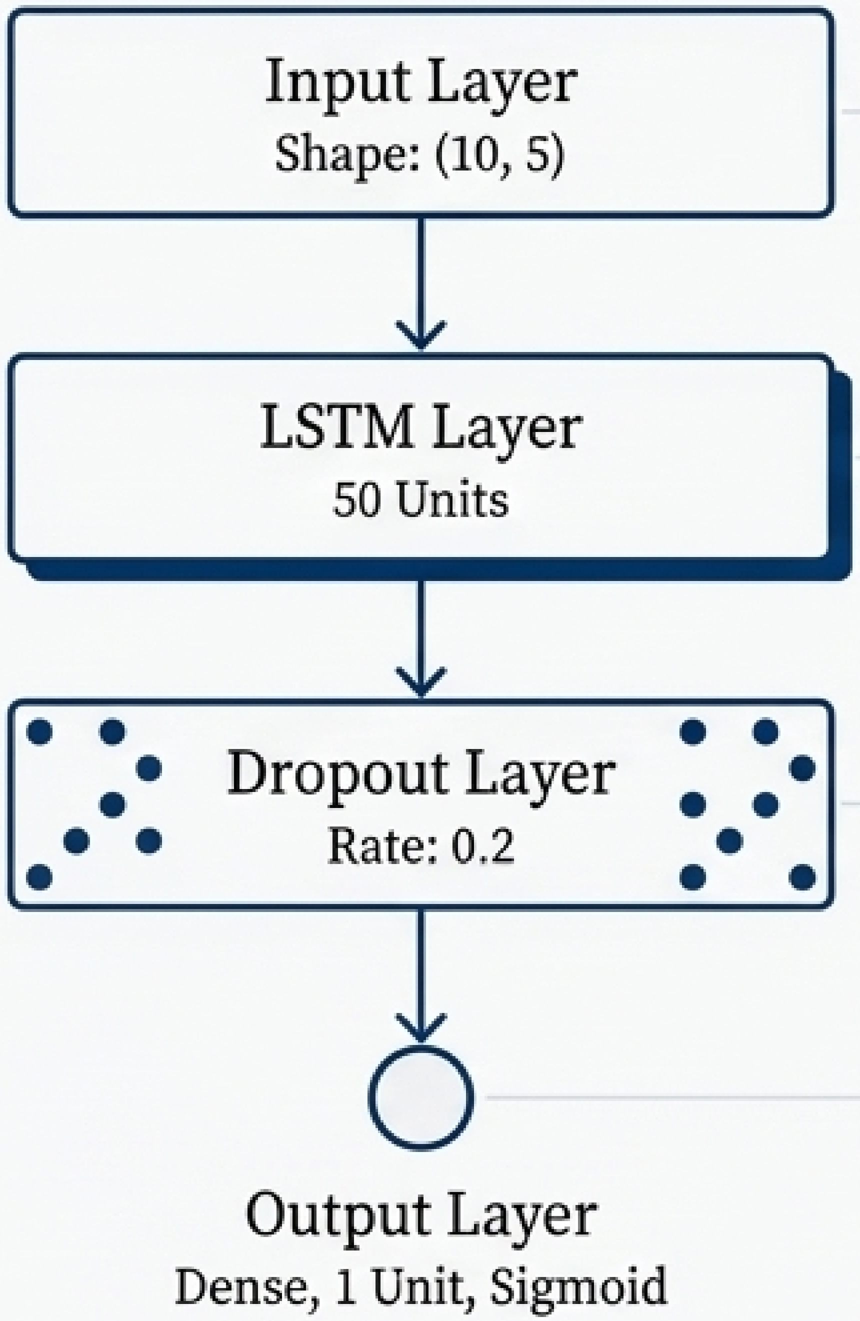
2. **Normalize Features**:All input features (OHLCV) were scaled to a range of [0, 1] using `MinMaxScaler`. This is critical for the stability and performance of LSTM networks.

The Model Was Taught to See Time in 10-Day Sequences

An LSTM model doesn't analyze a single day in isolation. It reviews a sequence of past data to make a prediction. We used a "lookback window" of 10 days of OHLCV data to predict the directional outcome of the 11th day.



We Designed a Lean LSTM Network to Learn from Market Patterns



10 days of 5 features (OHLCV).

Captures temporal dependencies and sequential patterns.

A regularization technique to prevent overfitting on the limited dataset.

Outputs a probability (0 to 1) of the price moving "Up".

```
model = Sequential()
model.add(LSTM(50, input_shape=(10, 5)))
model.add(Dropout(0.2))
model.add(Dense(1, activation='sigmoid'))
```

The Model Trained on the Past to Prepare for the Future

- **Data Split:** We used a strict chronological split to prevent data leakage. The first 80% of sequences were used for training, and the final 20% for testing.
- **Training Process:** The model was trained for 50 epochs using the Adam optimizer and binary cross-entropy loss function.

Direction of Time →

Chronological Split

Training Set (80%)

Held-Out Test Set (20%)

2024 dataset

The First Look at Performance: 61.7% Accuracy on the Test Set

61.7%

On the unseen test data, the model correctly predicted the daily price direction nearly 62% of the time. This performance is notably better than a random 50% chance, indicating the model learned a valid signal from the data.

But accuracy alone doesn't tell the full story.

The “Aha!” Moment Is Hidden in the Confusion Matrix

Model Predictions vs. Actual Outcomes

		Predicted Movement	
		Up	Down
Actual Movement	Up	3	18
	Down	0	26

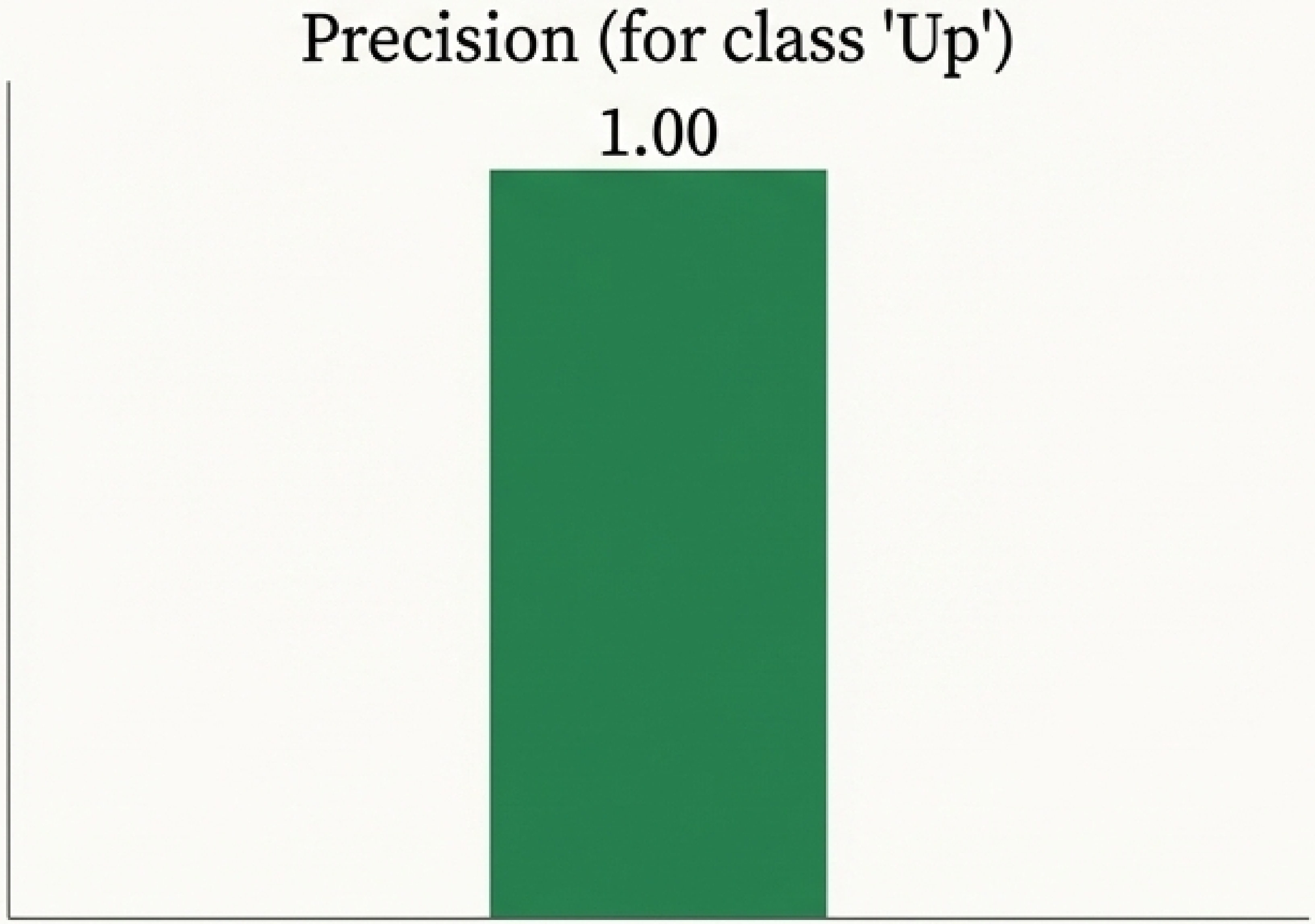
When it predicted "Up", it was right 100% of the time.

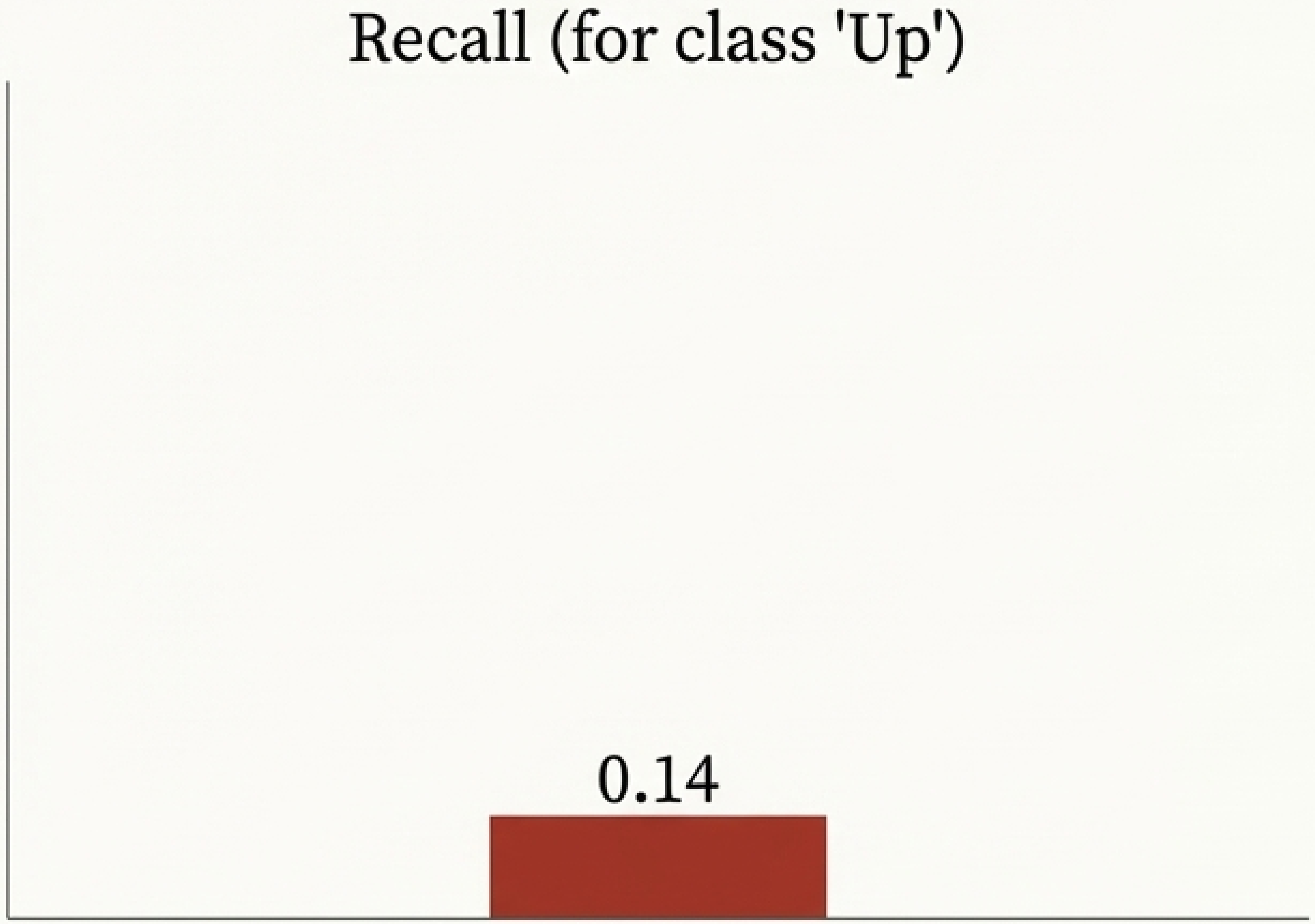
However, it missed the vast majority of actual "Up" days, predicting them as "Down".

It correctly identified a large number of "Down" movements.

Our Model Is a Cautious Specialist: Perfect Precision, Very Low Recall

The model adopted a highly conservative strategy. It only signals 'Up' when it has extreme confidence. This leads to perfect precision but means it overlooks most of the actual opportunities.





NotebookLM

The Model’s Conservatism Likely Stems from a Limited Historical View

With less than one year of training data, the model lacks exposure to a wide variety of market conditions. To minimize its primary objective (binary cross-entropy loss), it learned that predicting ‘Down’ is statistically the safer bet. The signal for ‘Up’ movements in the limited data was not strong or consistent enough for the model to risk being wrong.

This Is Not an Endpoint, But a Clear Roadmap for Improvement

The discovery of the model's conservative bias is the most valuable outcome of this case study. It provides a precise diagnosis of the current limitations and dictates the exact steps needed to build a more robust and confident forecasting model.

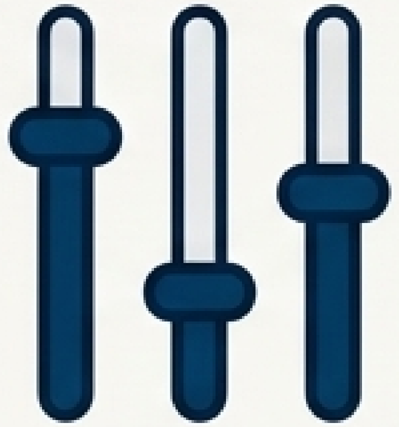
Three Key Levers to Enhance Predictive Power

1. Expand the Dataset

Increase the historical data range to 5–10 years. This will expose the model to more diverse market cycles and patterns, helping it build confidence.

2. Enrich the Features

Add contextual features beyond OHLCV. Incorporate common technical indicators like RSI, MACD, and Moving Averages to provide a richer signal.



3. Refine the Architecture

Systematically perform hyperparameter tuning. Experiment with the lookback window size, the number of LSTM units, and the optimizer's learning rate to find a more optimal configuration.

In Financial AI, Headline Accuracy Is Only the Start of the Story

Our journey to build a stock predictor began with a 61.7% accuracy figure. But the real value was not in the number itself, but in understanding the ***behavior*** behind it. Discovering our "Cautious Specialist" model tells us exactly how to move forward. True progress in data science comes from this deeper level of inquiry.