Predicting Amazon’s Stock Price

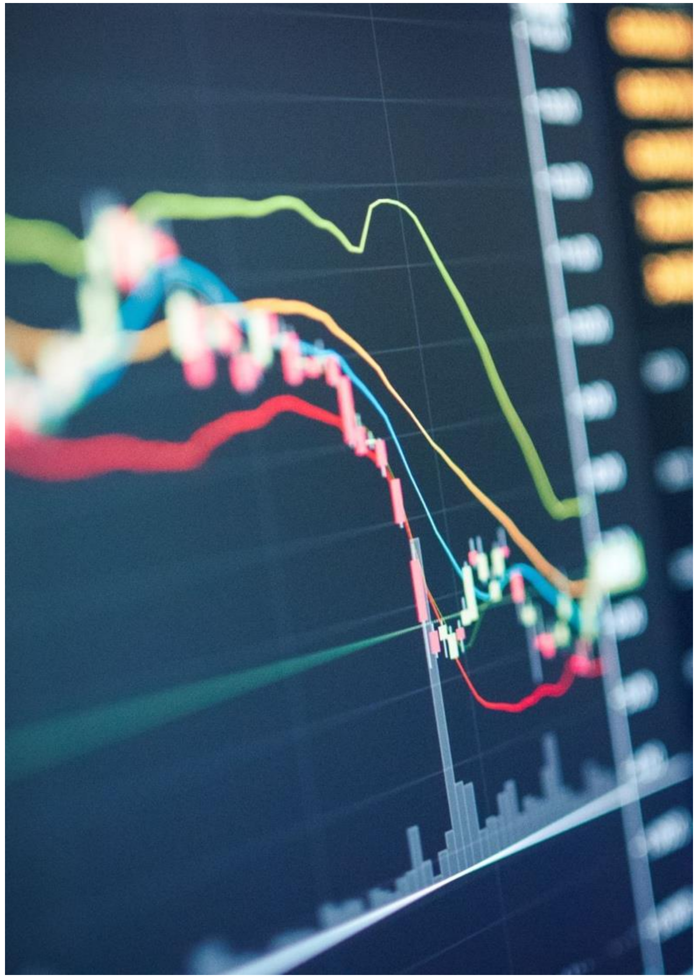
By Logan Pilcher

Outline

- Motivation
- Approach
- The Dataset and Visualizations
- LSTM Model
- Results and Evaluation
- Problems Encountered
- Key Observations

Motivation

- Stock market predictions are a popular topic now with the increasing popularity of AI and new deep-learning methods
- I wanted to combine my interest in the stock market and data science
- I chose Amazon mainly because of its popularity and because the dataset wasn't too big to work with, some popular stocks have been around for a long time like the S&P 500
- I wasn't sure how accurate this would be, but I wanted to experiment with Python methods and how successful they could be



My Approach

I researched and read different websites and papers on stock prediction in Python on the internet to get ideas for this project

The most common approach was using LSTM, so I opted to look more into this to focus on the LSTM and how to use it

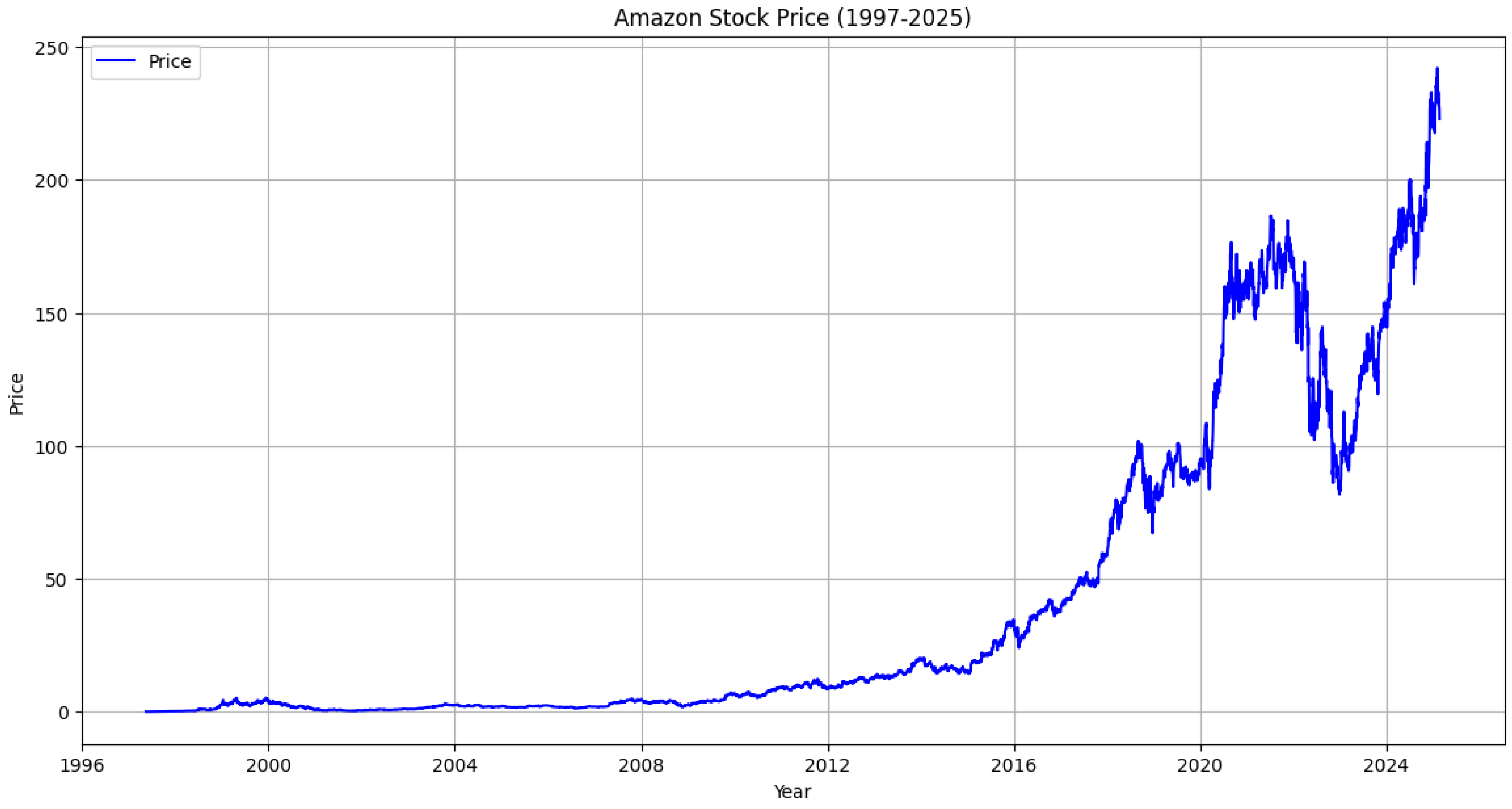
I originally planned on using multiple models, but I decided to focus only on LSTM and to make it as effective as possible, and predict the next 30-day stock price with this model

The Dataset

- It contains the price history of Amazon’s Stock from when it went public in 1997 to when I downloaded it a month ago (February 21st, 2025)
- **Date**: Daily stock prices over the period.
- **Open**: The opening price of the stock each day.
- **High/Low**: Highest and lowest prices recorded during the trading day.
- **Close**: The closing price of the stock each day.
- **Adjusted Close**: Closing price adjusted for dividends and stock splits (used for prediction).
- **Volume**: Number of shares traded each day.

index	date	open	high	low	close	adj_close	volume
0	1997-05-15 00:00:00-04:00	0.121875002980232	0.125	0.096354000270366	0.097916997969150	0.097916997969150	1443120000
1	1997-05-16 00:00:00-04:00	0.098438002169132	0.098958000540733	0.085417002439498	0.086457997560501	0.086457997560501	294000000
2	1997-05-19 00:00:00-04:00	0.088021002709865	0.088541999459266	0.081249997019767	0.085417002439498	0.085417002439498	122136000
3	1997-05-20 00:00:00-04:00	0.086457997560501	0.087499998509883	0.081771001219749	0.081771001219749	0.081771001219749	109344000
4	1997-05-21 00:00:00-04:00	0.081771001219749	0.082291997969150	0.068750001490116	0.071354001760482	0.071354001760482	377064000

Graph of Amazon's Price History (Until Feb 21, 2025)

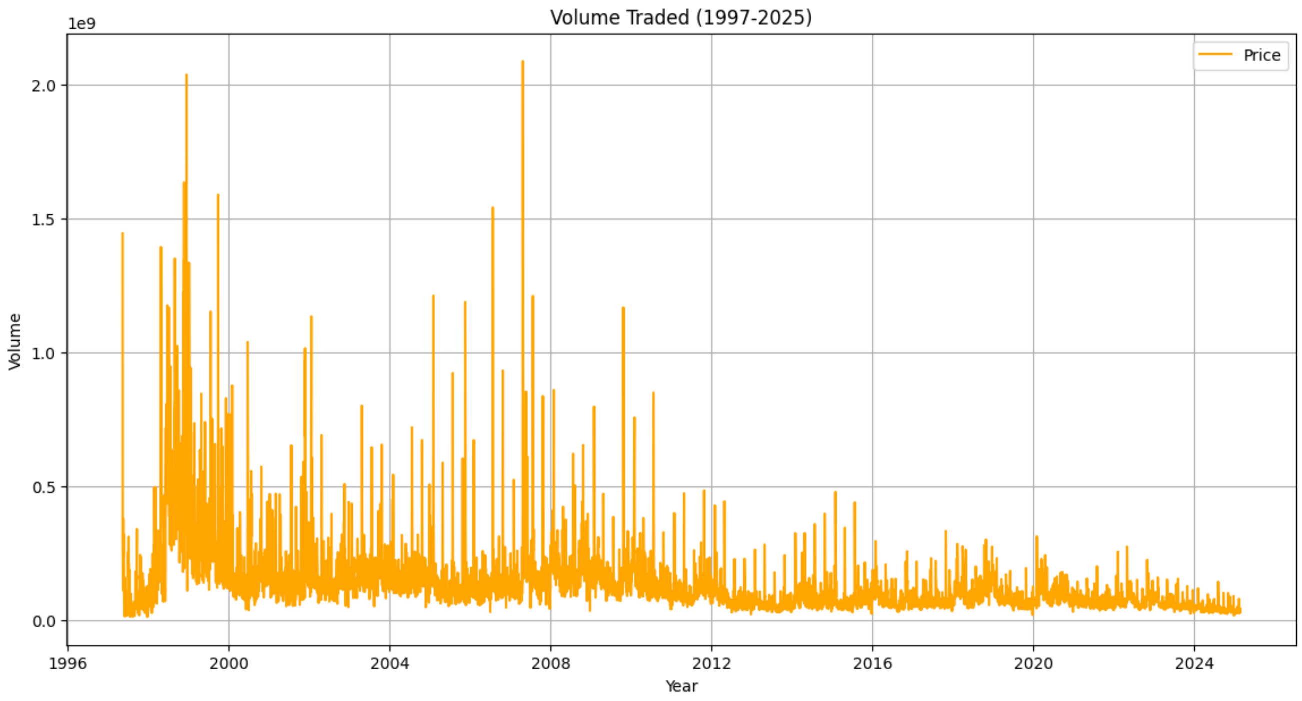


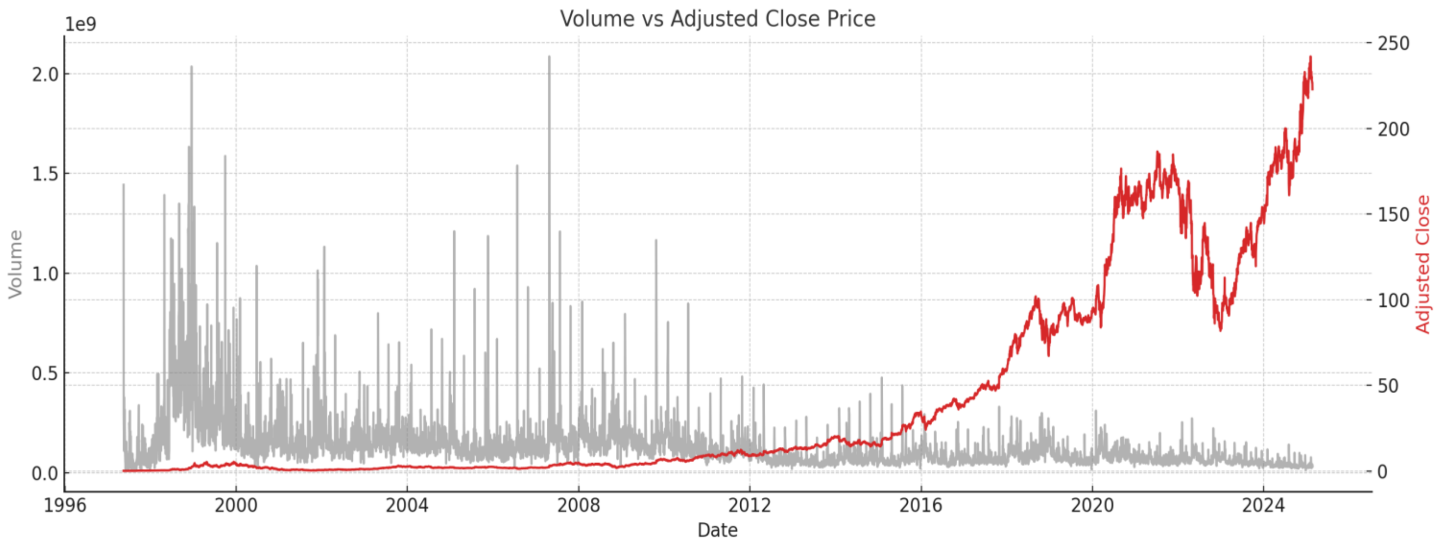
Graph of Volume Traded and Volume vs Adjusted Price

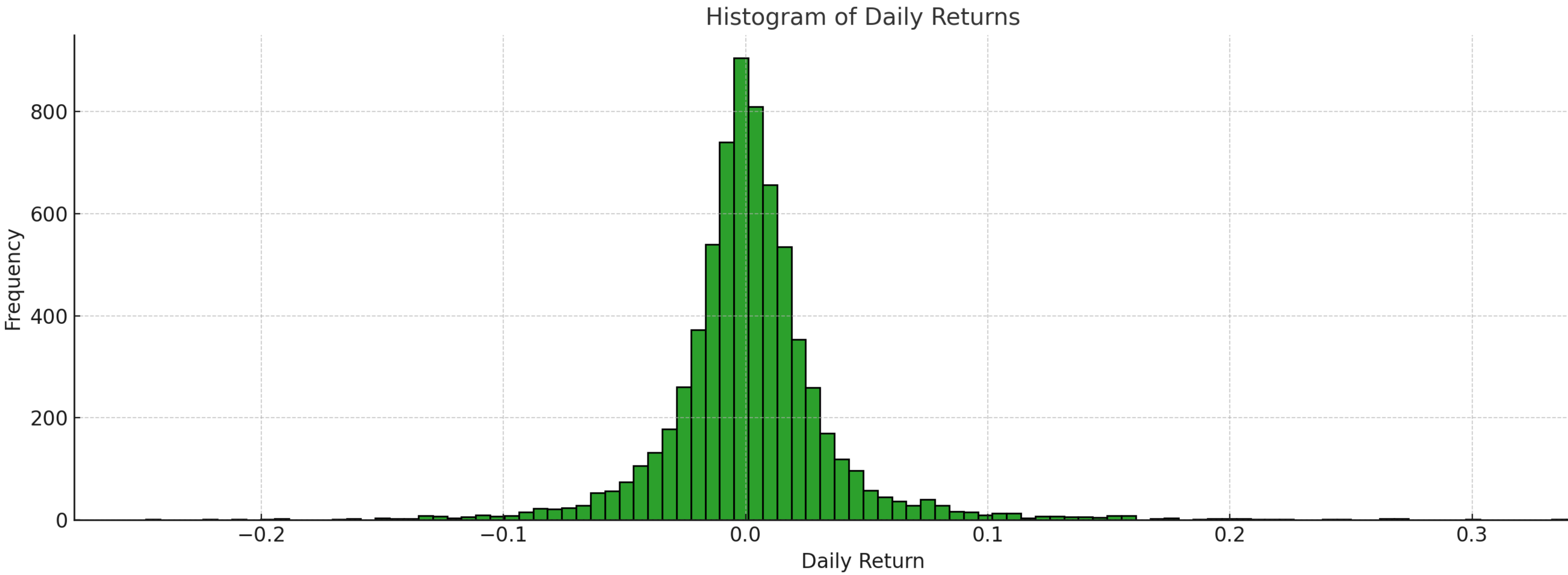
- The first graph shows the volume of Amazon's stock traded
- In recent years, it has shown both higher and more volatile trading activity, which could signal increased investor attention and market speculation.

Volume vs Adjusted Close Price Graph

- Compares trading volume (gray) with stock price (red).
- Spikes in volume often coincide with sharp price changes.
- Low volume can indicate uncertain or stagnant market sentiment
- Price jumps often occur after volume spikes → possible predictive signal
- Some volume-price movements are nonlinear, suggesting complex dynamics that my LSTM might need to learn.

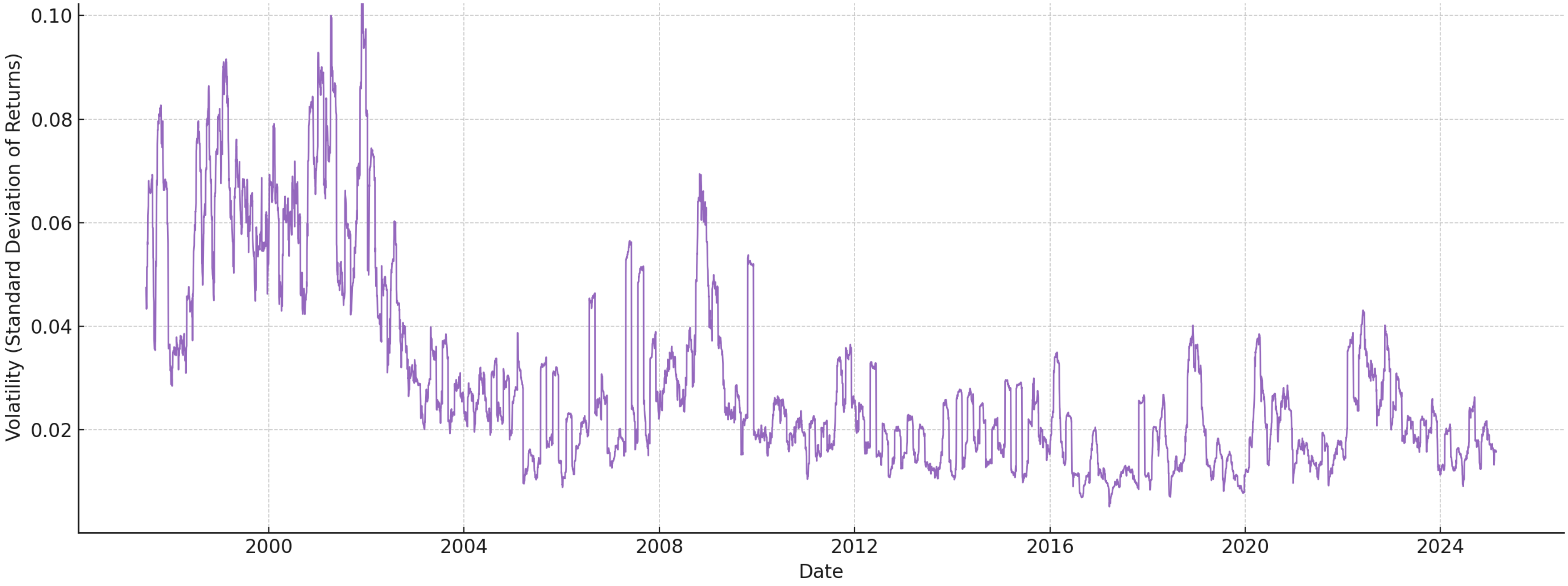






Histogram of Daily Returns

- Visualizes the distribution of Amazon's day-to-day return percentages.
- The graph suggests returns are roughly normally distributed.
- Most returns cluster around 0%, indicating that small daily movements are most common.
- The tails of the distribution (both positive and negative) show that extreme daily returns, while rare, do occur.
- This chart helps identify the frequency and magnitude of risk in the stock’s daily performance.



30-Day Rolling Volatility Graph

- This shows how the standard deviation of daily returns changes over time.
- Higher spikes indicate periods of greater uncertainty or market stress.
- Lower periods reflect market stability or investor confidence.
- Shows periods of calm vs. turbulence, which directly affects forecasting difficulty.

LSTM Model

LSTM (Long Short-Term Memory)

Type of Recurrent Neural Network (RNN) designed to handle long-term dependencies

Effective for time-series forecasting

Remembers patterns over long sequences while filtering out irrelevant data

Why LSTM for Stock Prediction?

Stock prices are time-series data – perfect for sequence-based models

LSTMs remember patterns from past trends, helping forecast future values

Can handle nonlinear and noisy patterns well

Capable of learning from multiple features (price, volume, indicators)

My LSTM Model

Preprocessing:

- Removed any missing values.
- Normalized features using MinMaxScaler.

Time Series Preparation:

- Created sequences of 60 days to predict the next day's price.
- For each time step of 60 days, predict the next day's price
- Split data: 80% training, 20% testing.

Model Architecture:

- LSTM layer with 100 units.
- Dropout layer to prevent overfitting.
- Dense output layer for price prediction.

Training Strategy:

- Used early stopping and learning rate reduction callbacks.
- Trained over 100 epochs with validation tracking.

Prediction & Evaluation:

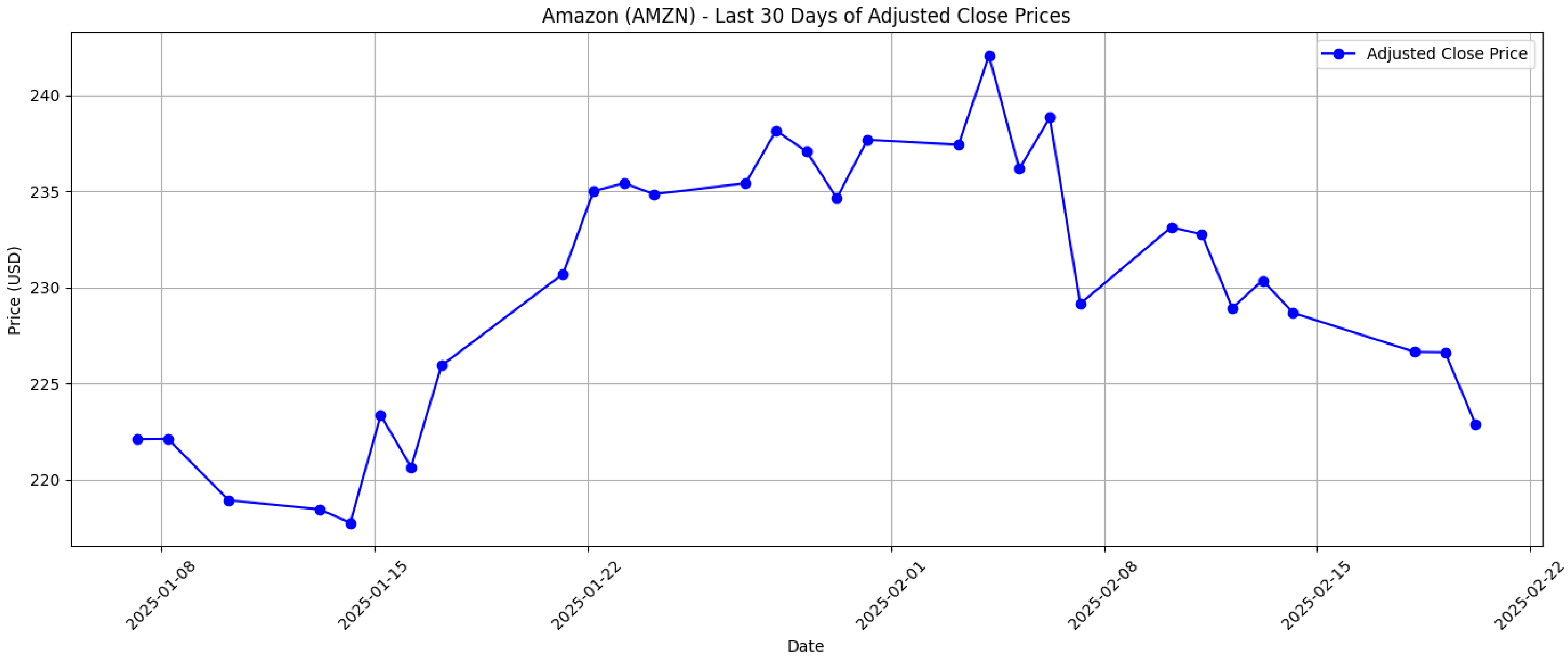
- Predicted prices on the test set.
- Inverted scaling to compare predictions with actual prices.

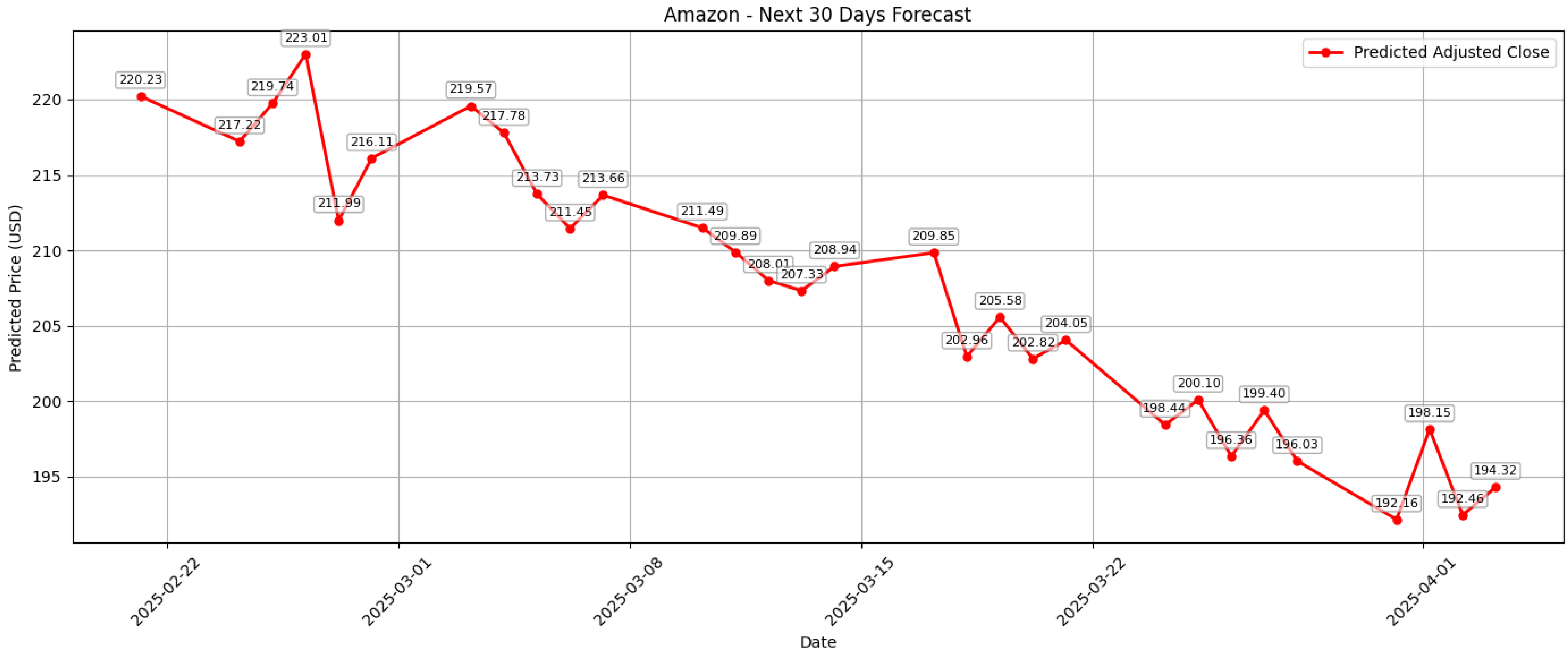
Results

LSTM Prediction Graph (Without 30-Day Prediction)

- The red prediction line is close to the actual stock movements, showing strong alignment and low error
- The model captures both upward trends and significant dips, demonstrating robust forecasting capability

Amazon's Last 30 Trading Days (Before February 21st)

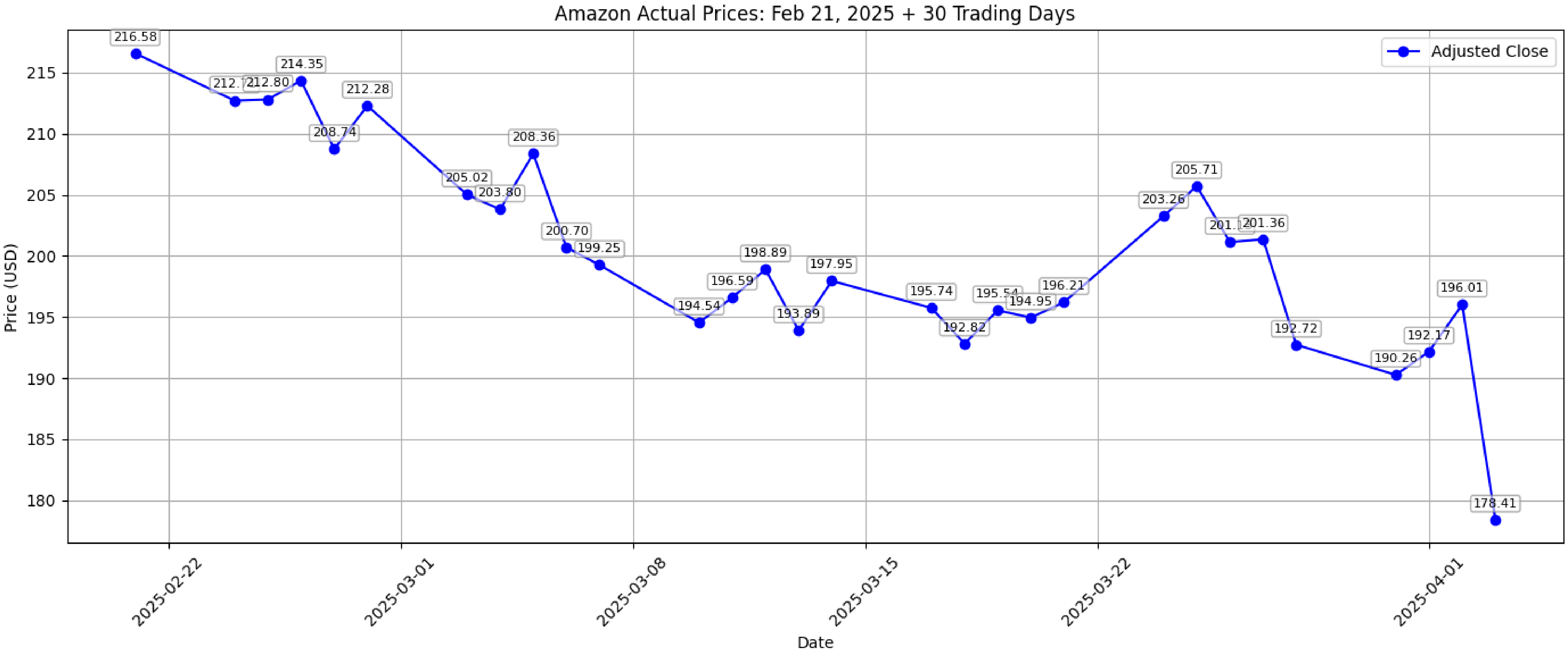




30-Day Predicted Price From February 21st

- This is the 30-day prediction from my LSTM Model
- Each point represents the model's predicted adjusted close price for a given future trading day.
- The model uses recent price momentum and volume trends to make the prediction
- Fairly linear descension

30 Day Actual Price from February 21st



The Prediction

Starts with recent data as input.

Predicts prices iteratively, day-by-day, updating inputs based on prior predictions.

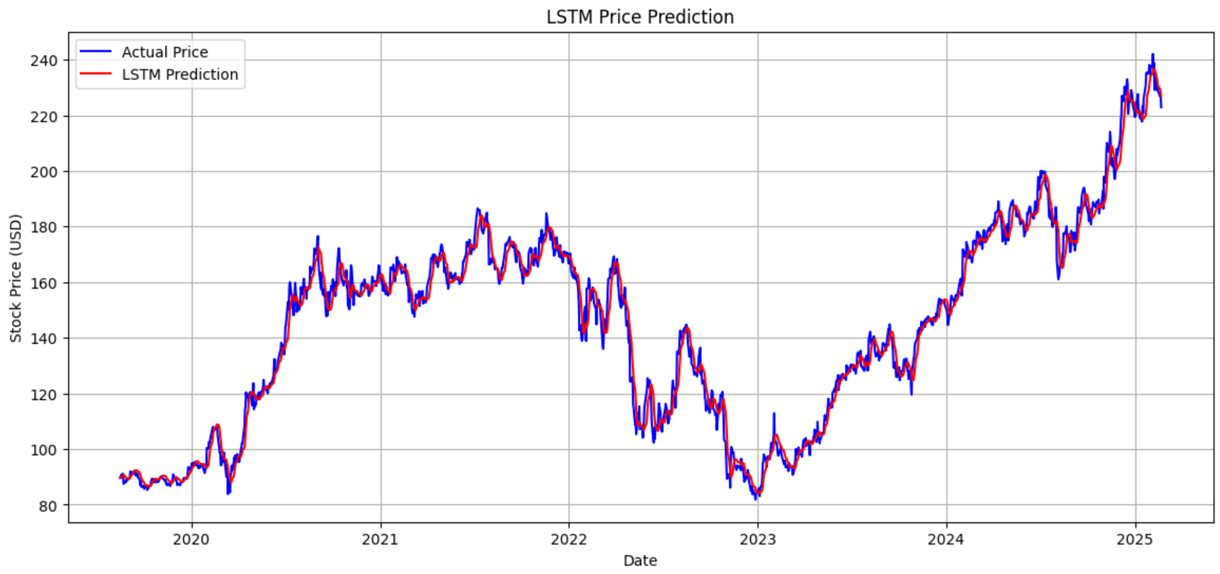
I added randomized daily volatility (±1%) to attempt to reflect real-world stock fluctuations.

It incorporates a downward drift bias (70% probability) to align with recent 30-day downward trend

Simulates realistic trading volumes with ±3% random noise.

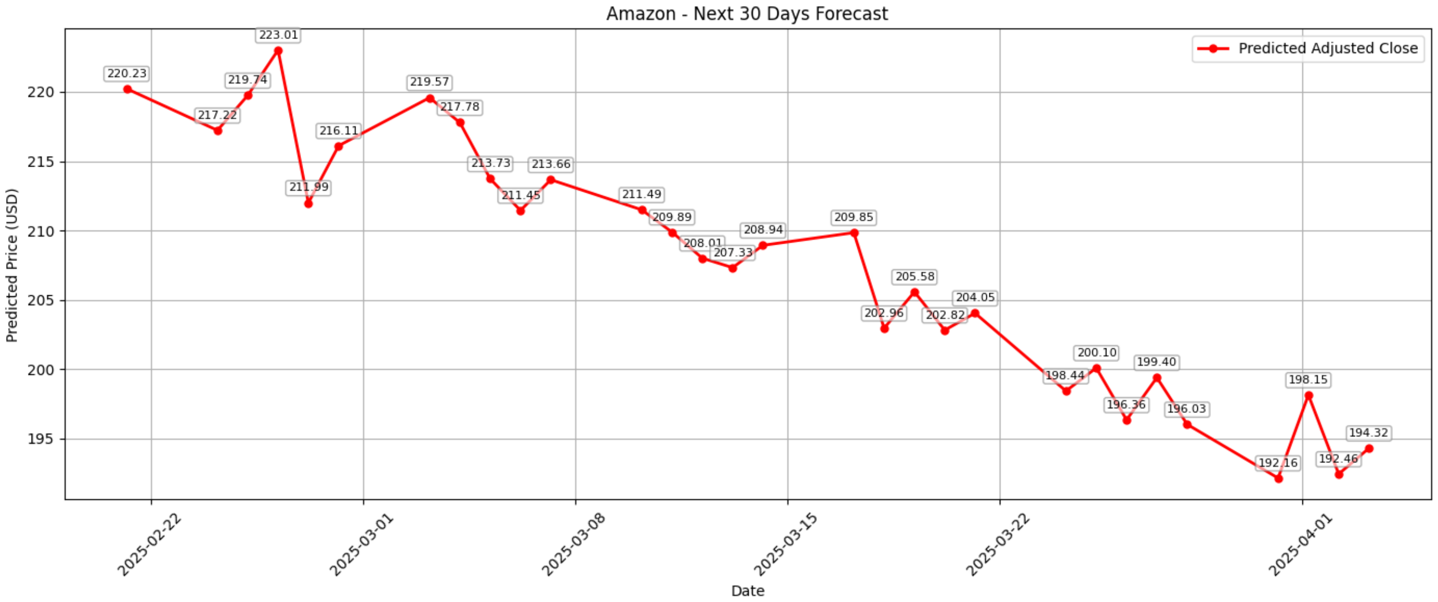
Evaluation

- My LSTM model learned long-term patterns: growth phases, dips, and trend reversals.
- The prediction line closely followed the actual price trajectory
- There is a slight lag on sharp price drops or spikes, which is typical of LSTM's temporal smoothing.
- The model captured momentum, volatility cycles, and price stabilization zones.
- No significant overfitting signs, generalization remained consistent in validation data.



30-Day Prediction Evaluation

- Predicted Range: ~$220.23 to ~$192.46
- Actual Range: ~$216.58 to ~$178.41
- My prediction did not anticipate the depth of the final price drop, underestimating the volatility and end-point decline
- The actual prices bounced up and down more irregularly, suggesting reactivity to short-term events that were not part of your model inputs
- Incorporated volatility and directional bias into the LSTM loop
- Matched the macro shape and sentiment of the market reasonably well without overfitting
- The forecast stayed within a reasonable range of the actual adjusted close prices
- The predicted trend demonstrated market volatility, but with more inaccurate values



Problems Encountered

Using machine learning for this predicting stock prices does not come without its challenges

LSTM models rely on past data, but markets react to real-time news, earnings, and sentiment; this can't be taken into consideration for predicting, context matters

Underestimating volatility, my LSTM tends to smooth out rapid fluctuations, making it less responsive to sudden market changes.

Single-output limitation, forecasting one value at a time, can lead to cumulative error in predictions such as my 30-day outlook

Key Observations

I was surprised with my results; I didn't expect them to be very accurate, but my models looked pretty good and were relatively close

The model performed well in identifying overall trends, showing strong alignment with the actual price direction

Limitations observed in short-term volatility capture and responsiveness to real-time market events.

I gained a better understanding of how stock prices reflect more than data points

Predicting the market is a complex, imperfect task, but can offer valuable insight into market behavior.

I had a lot of fun writing this up, it was a rewarding project to see my work and visualize all that I did