Predicting the Stock Price of Amazon using LSTM

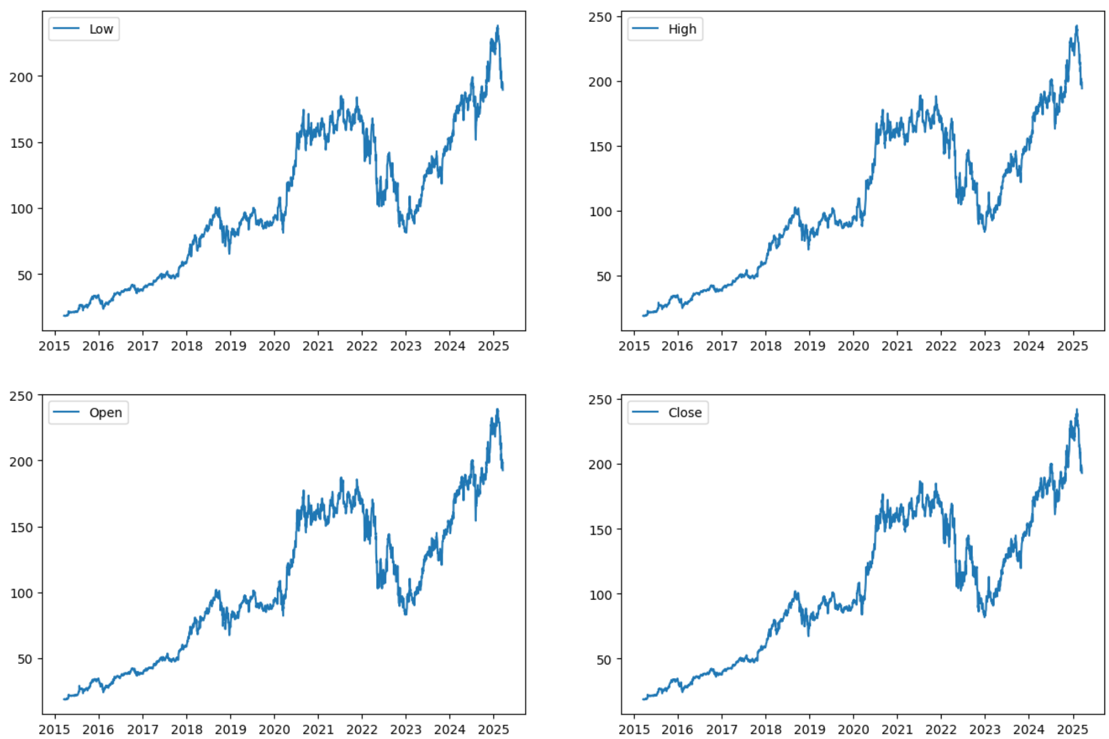
Introduction
Stock market price prediction is a challenging task due to its complex and volatile nature. Traditional models often struggle with capturing time-series dependencies, making deep learning techniques like Long Short-Term Memory (LSTM) networks highly effective for forecasting stock prices. This project leverages an LSTM model to predict future stock prices based on historical data.

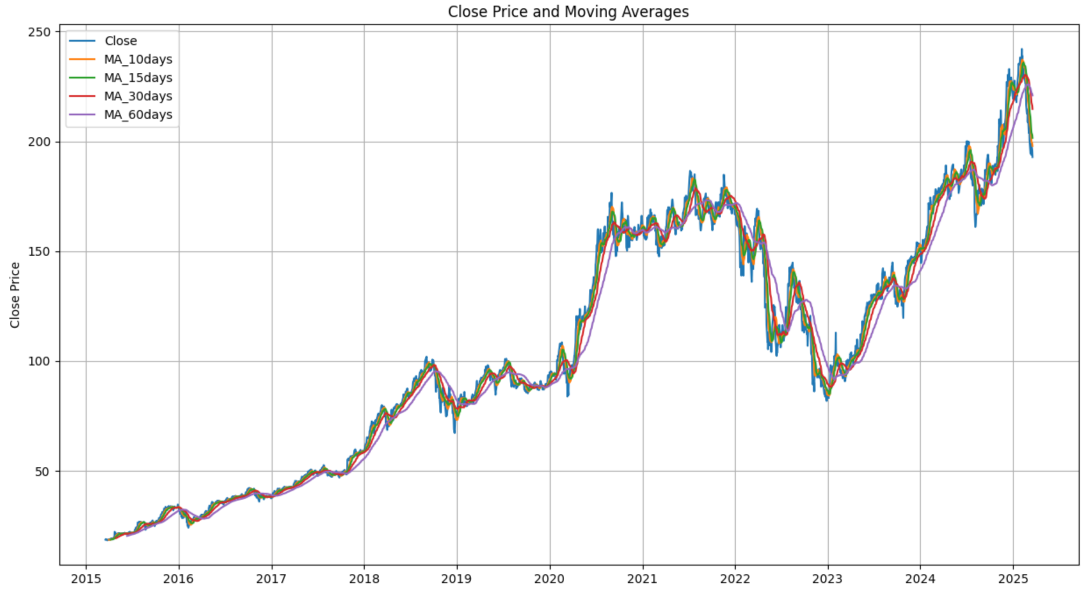
Objective
The primary goal of this project is to develop an LSTM-based predictive model that forecasts stock close prices for the next n days using past trading data. The model is trained to learn temporal dependencies and trends in stock prices to make accurate future predictions.

Data Source:
10 year historical stock market data consisting of Open, High, Low, Close, and Volume features. Extracted from Yahoo Finance or other stock market APIs.

Data Processing:
1. Calculating the 10, 15, 30 and 60 day Moving Average
2. Calculating Spread = High - Low
3. Feature scaling using MinMaxScaler to normalise between 0 and 1
4. Creating a sequence of 60 past stock prices to train the LSTM model
5. Splitting the data into train (80%), validation (10%) and test (10%) sets

Explanatory Data Analysis:





Model Training and Evaluation

The LSTM model consists of the following layers:

- Input Layer: Takes in sequences of historical stock prices.
- LSTM Layers: Three stacked LSTM layers with dropout regularization to prevent overfitting.
- Dense Layer: Fully connected layer to output the predicted stock price.
- Activation Function: ReLU for hidden layers and `Linear` for the output layer.
- Loss Function: Mean Squared Error (MSE) to minimize the error in predictions.
- Optimizer: Adam optimizer for efficient gradient updates.

Layer (type)	Output Shape	Param #
lstm_3 (LSTM)	(None, 60, 128)	68,608
dropout_2 (Dropout)	(None, 60, 128)	0
lstm_4 (LSTM)	(None, 60, 128)	131,584
dropout_3 (Dropout)	(None, 60, 128)	0
lstm_5 (LSTM)	(None, 64)	49,408
dense_2 (Dense)	(None, 32)	2,080
dense_3 (Dense)	(None, 1)	33

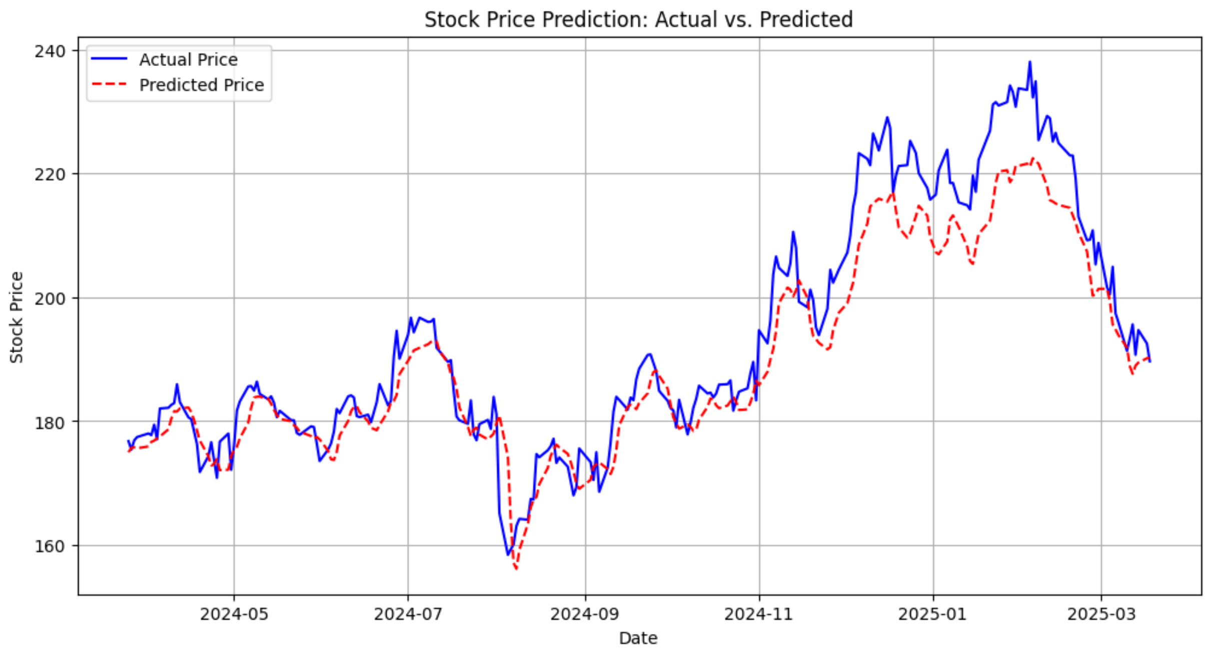
Total params: 251,713 (983.25 KB)
Trainable params: 251,713 (983.25 KB)
Non-trainable params: 0 (0.00 B)

To optimize training, we incorporated **callbacks** that help in improving generalization and avoiding overfitting:
Early Stopping prevents overfitting by stopping training when the validation loss stops improving for 10 consecutive epochs. It helps the model converge better by adjusting learning rates dynamically.
Learning Rate Scheduler reduces the learning rate by a factor of 0.8 if validation loss does not improve for 5 epochs. It helps the model converge better by adjusting learning rates dynamically.

The LSTM model is trained using the training dataset with a batch size of 32 and trained for 100 epochs.
The validation monitors overfitting and model performance.

The model's performance is evaluated using:
- Mean Absolute Error (MAE)
- Mean Squared Error (MSE)

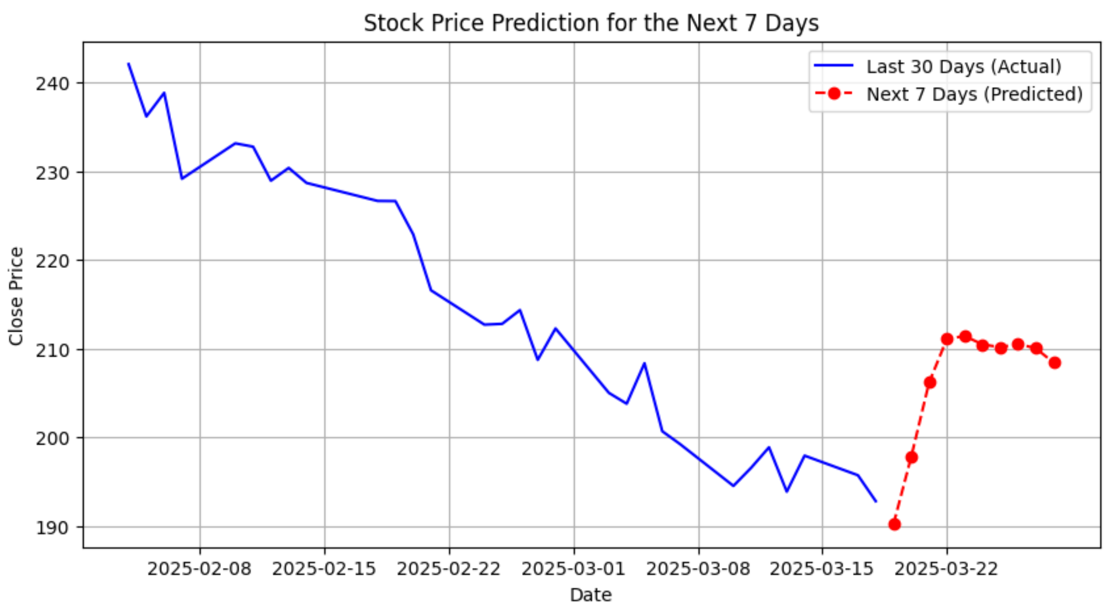
Results and Visualization



The actual vs predicted stock prices are plotted to visualize model performance.
Training Loss: 0.000503
Validation Loss: 0.00085
MAE on Test set: 0.0221

Future Price Prediction

A function is implemented to predict future stock prices for the next n days based on the last available sequence. The process involves:
1. Using the trained LSTM model to predict the next day's stock price.
2. Updating the input sequence with the new prediction.
3. Iteratively predicting for n future days.
4. Applying inverse transformation to convert predictions back to original scale.



	Date	Predicted Close Price
0	2025-03-19	190.25
1	2025-03-20	197.86
2	2025-03-21	206.20
3	2025-03-22	211.11
4	2025-03-23	211.42
5	2025-03-24	210.49
6	2025-03-25	210.16

Limitations:

1. Stock prices are influenced by external factors (news, earnings reports, global events) that are not included in the dataset.
2. Despite regularization, LSTMs can overfit on small datasets.
3. Training deep learning models on large datasets requires high computational power.