Project Title: Stock Market Analysis and Forecasting Using Deep Learning
Student: Emily Liang
Course: COE 379L

Introduction and Problem Statement
I always felt that the stock market is a complex and dynamic system influenced by numerous factors, making accurate predictions difficult. Traditional methods often cannot capture the intricate patterns and non-linear relationships that are inherent in financial data. Therefore, I believe that with deep learning, it can be leveraged with these advanced algorithms to analyze historical stock data and accurately predict future trends better.

This project aims to utilize deep learning techniques, specifically Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks, to analyze and predict stock prices. By focusing on historical data from major companies, the goal is to develop models that can accurately predict what the stock will do.

Data Sources
Historical stock data was obtained for the following companies:

- Google (GOOG) – 2006–2024
- Microsoft (MSFT) – 2006–2024
- IBM (IBM) – 2006–2024
- Amazon (AMZN) – 2006–2024
- Meta (META, used to be Facebook) – 2016–2024
- Apple (AAPL) – 2006–2024
- Netflix (NFLX) – 2006–2024
- Tesla (TSLA) – 2010–2024

Each dataset includes the data with the following attributes:

- Open Price
- High Price
- Low Price
- Close Price
- Volume

Originally, I planned on getting the dataset from a dataset website like Kaggle or Hugging Face, however, I found the *yfinance* Python package that collects the historical stock data using the Yahoo Finance API.

Technologies Used
For this project, I used:

- *NumPy* and *Pandas* for data processing
- *Matplotlib* for visualization
- *scikit-learn* for data scaling
- *TensorFlow/Keras* for building and training deep learning models

The model that I am specifically using is LSTM…

Methods Employed

The model I am using for this model is the LSTM, which stands for Long Short-Term Memory. It is a type of Recurrent Neural Network (RNN). We learned about RNNs in class, and they are a class of neural networks designed to work with sequential data.

However, traditional RNNs suffer from the vanishing gradient problem because they struggle to learn long-term dependencies (like trends across months or years in stock prices). LSTMs solve this problem using a special architecture that allows them to retain important information over long sequences and forget irrelevant details.

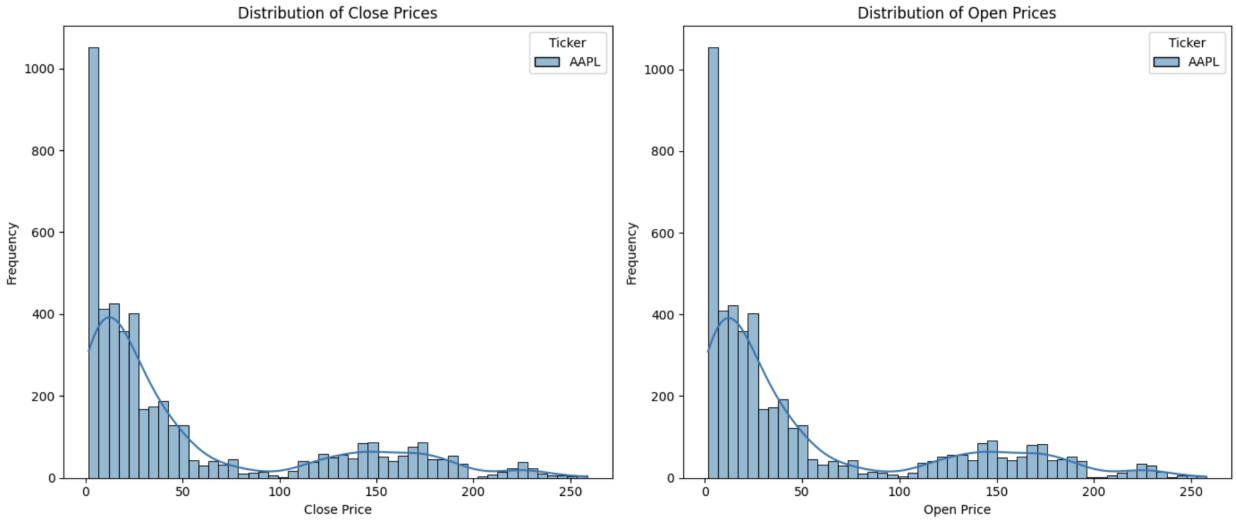
LSTM for stock prices is better since stock prices are sequential, since yesterday's price influences today's price, and are noisy but with patterns such as trends or seasonality. LSTMs are good for this type of task because they can learn long-term dependencies (for this model, 60-day windows), avoid short-term noise and focus on trend signals, and retain memory of prior days in a structured way.

Before doing anything, the dataset needs to be analyzed. Note: The images below are for Apple; the rest of the graphs are in the notebook.

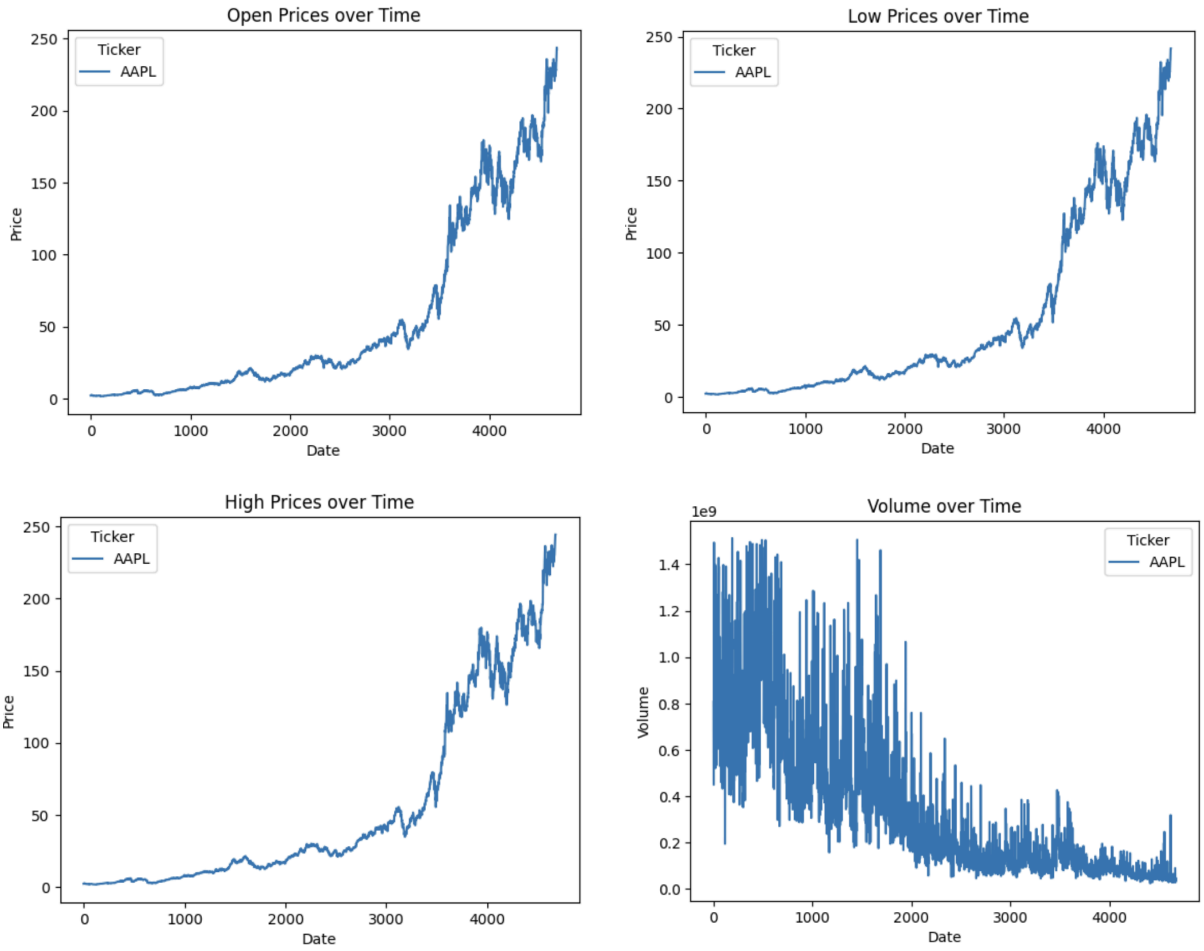
- Describe:

Price	Close	High	Low	Open	Volume
Ticker	AAPL	AAPL	AAPL	AAPL	AAPL
Date					
2006-01-03	2.249497	2.249497	2.174263	2.178175	807234400
2006-01-04	2.256118	2.286512	2.241974	2.260933	619603600
2006-01-05	2.238363	2.254011	2.219404	2.251905	449422400
2006-01-06	2.296142	2.308179	2.243478	2.264543	704457600
2006-01-09	2.288618	2.323226	2.279289	2.309082	675040800
...	...	...	...	...	...
2024-12-23	254.989655	255.369227	253.171646	254.490204	40858800
2024-12-24	257.916443	257.926411	255.009620	255.209412	23234700
2024-12-26	258.735504	259.814335	257.347047	257.906429	27237100
2024-12-27	255.309296	258.415896	252.782075	257.546826	42355300
2024-12-30	251.923019	253.221595	250.474615	251.952985	35557500

- Distribution of Close and Open



- Visualize the attributes[Open, High, Low, Close, volume] of our datasets.



- Trend and seasonality in the dataset

Before the model can be trained, the data needs to be preprocessed.

- **Normalization**: Scales the stock price data to a range between 0 and 1. LSTM models work best when input data is normalized or standardized. Stock prices can vary widely, and normalization prevents the model from being biased toward large values.
- **Sequencing**: For each company, a window of 60 days of past data was used to predict the next day's price. Since LSTM models expect input in the shape (samples, timesteps, features), here, I made it so that it is what the model expects.

LSTM Model Architecture:

- Two LSTM layers with 50 units each
- One fully connected dense layer with 1 output node
- Mean squared error as the loss function
- Adam optimizer for training

Training:

- Epochs: 10

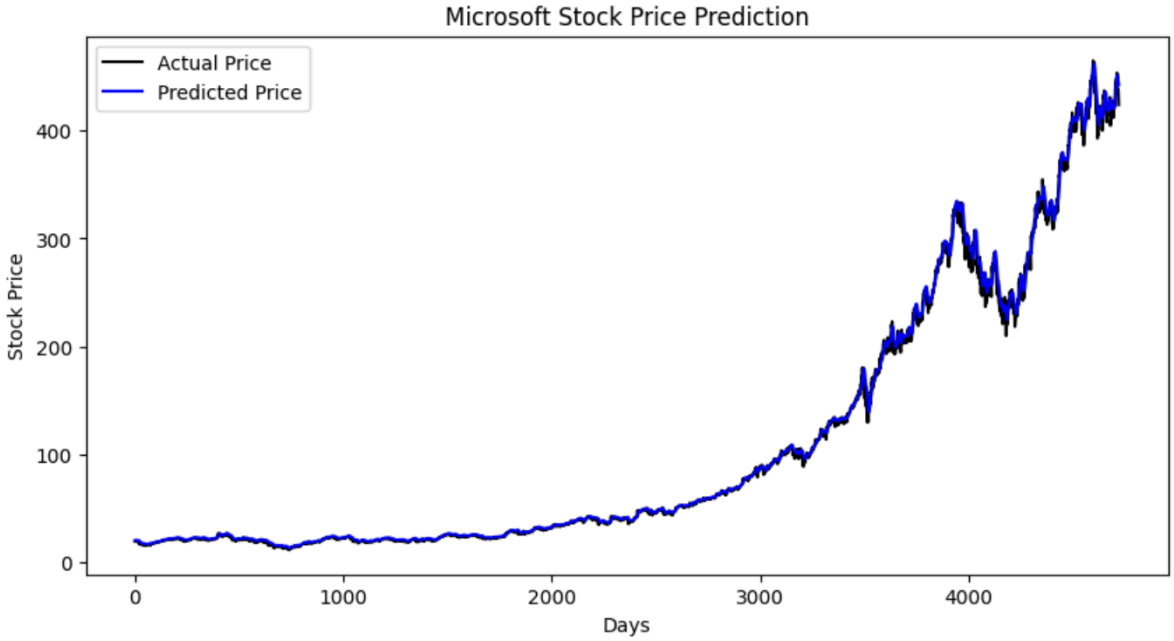
- Batch Size: 32

Note: I originally was going to also use RNN; however, after comparing it to the LSTM, it performed significantly worse in many aspects.

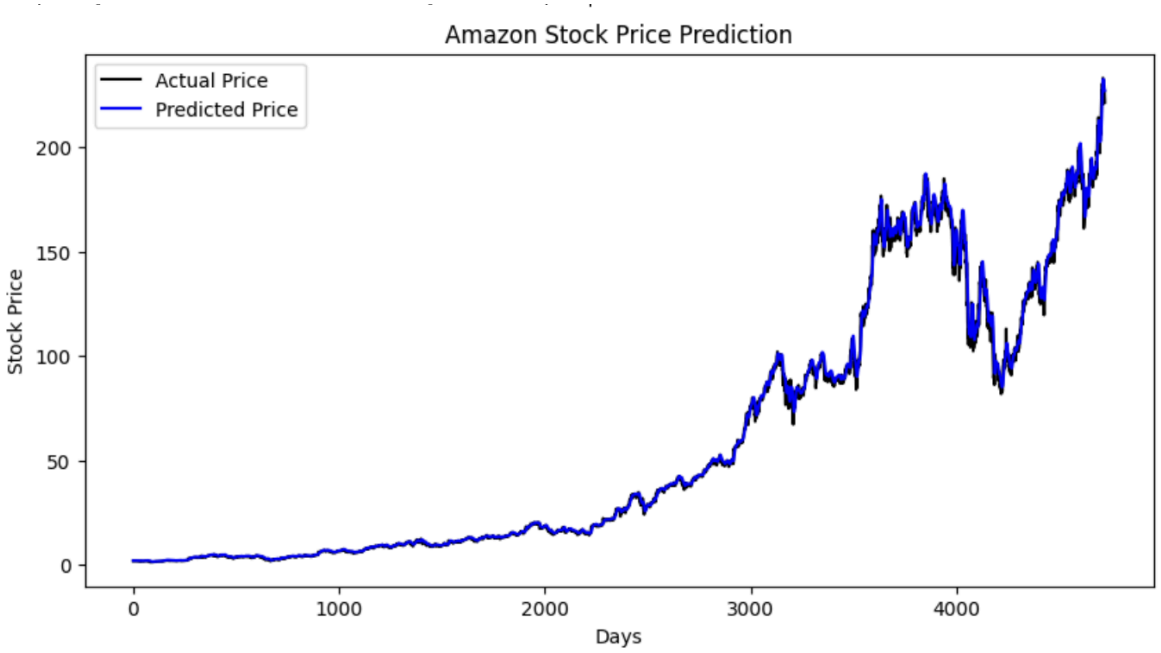
Results

Visual inspection was used to compare model predictions against actual stock prices. The LSTM was able to track general trends but often failed to capture extreme price movements or sudden volatility spikes.

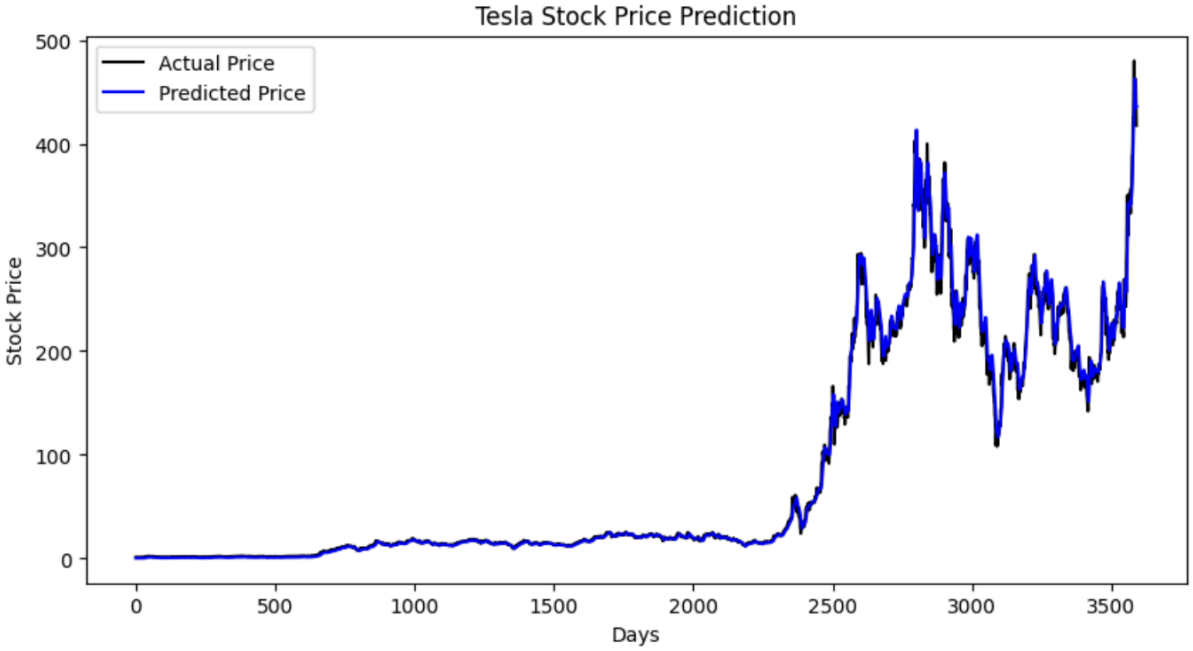
Sample Outcomes



Predicted prices followed long-term trends reasonably well, but it did not have great precision in short-term fluctuations.



LSTM captured the upward trend but lagged behind rapid growth phases.



Higher volatility led to noisier predictions; the model underfitted extreme variations.

Some strengths of the model are that general upward or downward trends were learned and followed, and it is good at smooth, trend-like data. However, it has weaknesses such as being

sensitive to overfitting on smaller datasets and not taking into account of current political and economic context.

Some key takeaways for me were that LSTM models are useful for approximating stock market behavior over time but may need to be combined with external features (e.g., sentiment analysis, news, macroeconomic indicators) for real-world trading applications. Overall, the model is a good thing to use to get a feel for historical data and what historical data says will happen in the future. However, stocks often do not follow historical trends and are affected by different factors. In addition, performance can vary greatly depending on the company's volatility and history length.

References

ChatGPT: Explanations for LSTM and how stocks work. It helped me debug some bugs that I had while cleaning data. I was trying to produce graphs, but I did not realize I had missed some steps in cleaning the dataset.
Yfiance: https://pypi.org/project/yfinance/
LSTM: https://colah.github.io/posts/2015-08-Understanding-LSTMs/
https://www.geeksforgeeks.org/deep-learning-introduction-to-long-short-term-memory/
Stocks: https://www.schwab.com/stocks/understand-stocks

Video Presentation
https://www.loom.com/share/26edb283c53140cf907350859f05895f?sid=9f28a61e-6c44-4ba2-8792-400ca8e8a5da

Presentation
https://docs.google.com/presentation/d/1eJOdC0Q-Ut1mhq9DLkYFRDr5B9T9fLEM4BKZEjgjZ1c/edit?usp=sharing