Min-Soo Mun (mxm154930)

CS 6375.003

Final Project Report

Topic: Google Stock Price Prediction Using LSTM

Dataset: https://www.kaggle.com/datasets/medharawat/google-stock-price?resource=download&select=Google_Stock_Price_Train.csv

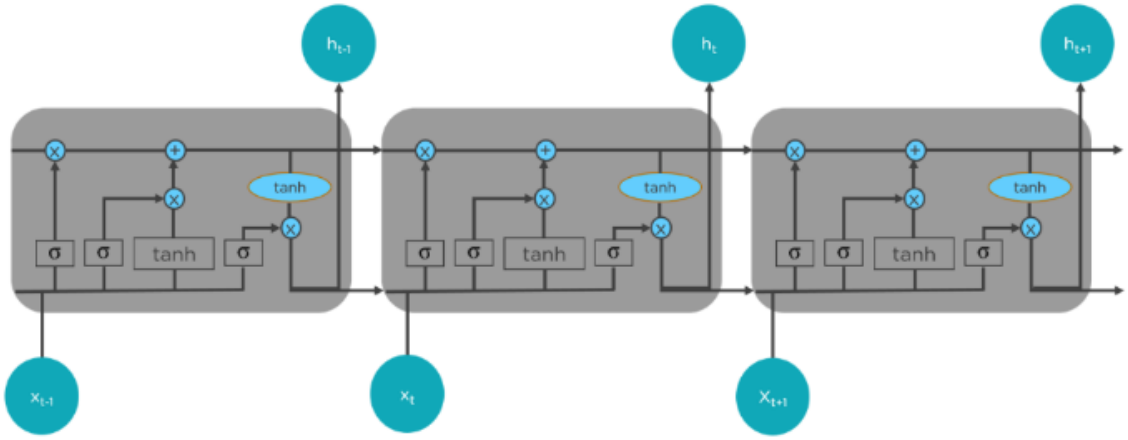
Introduction:

In this project, I aim to predict the stock prices of Google using a deep learning model called Long Short-Term Memory (LSTM). The prediction of stock prices has always been an important area of research in finance and economics. Stock prices are highly volatile and depend on various factors such as the company's financial performance, global events, market trends, etc. Accurately predicting the stock prices can help investors make informed decisions about buying and selling stocks, minimizing risks and maximizing profits.

Long Short Term Memory Network (LSTM):

I used a Long Short Term Memory Network (LSTM) for building my model to predict the stock prices of Google.

LSTMs are a type of Recurrent Neural Network for learning long-term dependencies. It is commonly used for processing and predicting time-series data.



From the image above, you can see LSTMs have a chain-like structure. General RNNs have a single neural network layer. LSTMs, on the other hand, have four interacting layers communicating extraordinarily.

LSTMs work in a three-step process.

- The first step in LSTM is to decide which information to be omitted from the cell in that particular time step. It is decided with the help of a sigmoid function. It looks at the previous state (ht-1) and the current input xt and computes the function.
- There are two functions in the second layer. The first is the sigmoid function, and the second is the tanh function. The sigmoid function decides which values to let through (0 or 1). The tanh function gives the weightage to the values passed, deciding their level of importance from -1 to 1.
- The third step is to decide what will be the final output. First, you need to run a sigmoid layer which determines what parts of the cell state make it to the output. Then, you must put the cell state through the tanh function to push the values between -1 and 1 and multiply it by the output of the sigmoid gate.

Data Collection:

I used the Google training data from 3 Jan 2012 to 30 Dec 2016.

	Date	Open	High	Low	Close	Volume
0	1/3/2012	325.25	332.83	324.97	663.59	7,380,500
1	1/4/2012	331.27	333.87	329.08	666.45	5,749,400
2	1/5/2012	329.83	330.75	326.89	657.21	6,590,300
3	1/6/2012	328.34	328.77	323.68	648.24	5,405,900
4	1/9/2012	322.04	322.29	309.46	620.76	11,688,800

There are five columns. The Open column tells the price at which a stock started trading when the market opened on a particular day. The Close column refers to the price of an individual stock when the stock exchange closed the market for the day. The High column depicts the highest price at which a stock traded during a period. The Low column tells the lowest price of the period. Volume is the total amount of trading activity during a period of time.

Data Preprocessing:

I extracted the closing prices as my target variable and normalized it using MinMaxScaler.

Normalization helps in scaling the values between 0 and 1 and reduces the impact of outliers on the model's performance.

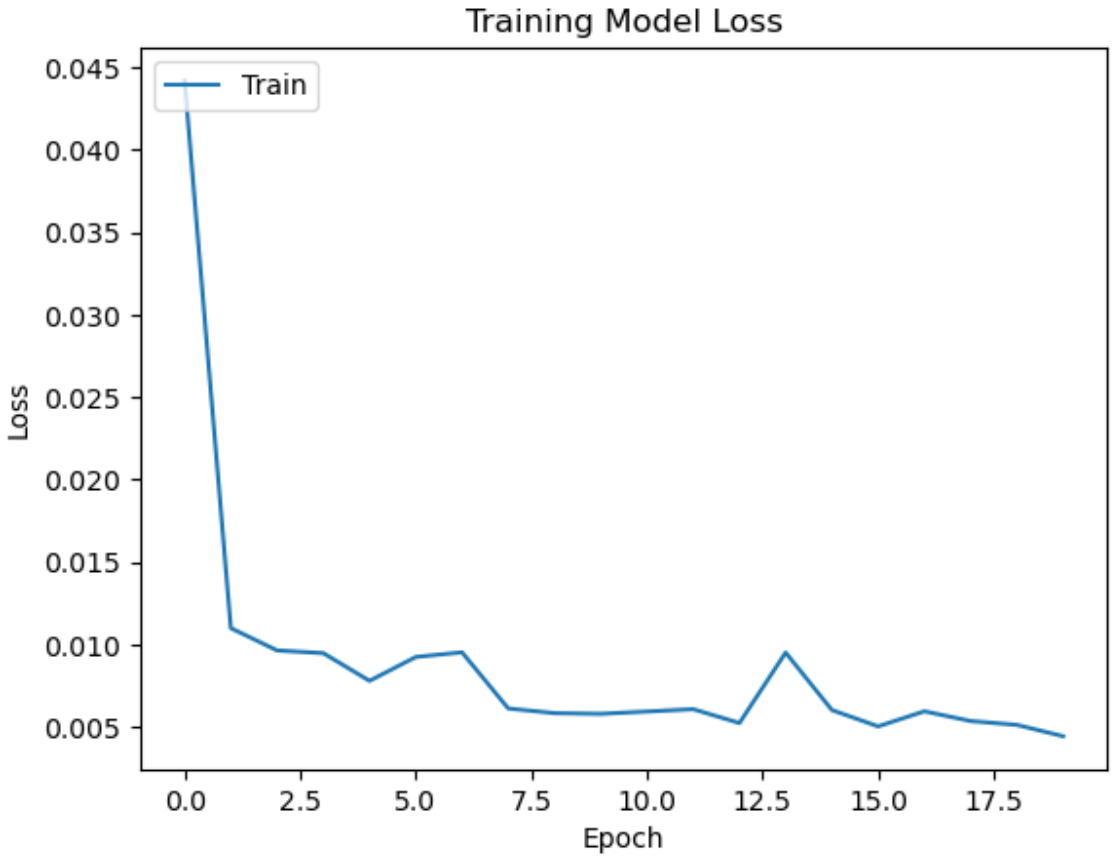
Then, I split the dataset into training and testing sets. I used the first 60 days of the data as the input sequence for the LSTM model and predicted the closing price for the 61st day. I repeated this process for the entire dataset, generating a total of 1149 samples for training. For testing, I used the same procedure on the remaining data.

Model Architecture:

I used a sequential LSTM model for my prediction task. The model consisted of four LSTM layers, each followed by a dropout layer to avoid overfitting. I used the "adam" optimizer and "mean squared error" loss function to train the model. I trained the model for 20 epochs with a batch size of 32.

```
Epoch 1/20
35/35 - 23s - loss: 0.0488 - 23s/epoch - 651ms/step
Epoch 2/20
35/35 - 8s - loss: 0.0125 - 8s/epoch - 237ms/step
Epoch 3/20
35/35 - 8s - loss: 0.0101 - 8s/epoch - 235ms/step
Epoch 4/20
35/35 - 8s - loss: 0.0102 - 8s/epoch - 233ms/step
Epoch 5/20
35/35 - 8s - loss: 0.0096 - 8s/epoch - 238ms/step
Epoch 6/20
35/35 - 9s - loss: 0.0077 - 9s/epoch - 249ms/step
Epoch 7/20
35/35 - 9s - loss: 0.0073 - 9s/epoch - 243ms/step
Epoch 8/20
35/35 - 9s - loss: 0.0082 - 9s/epoch - 248ms/step
Epoch 9/20
35/35 - 9s - loss: 0.0074 - 9s/epoch - 251ms/step
Epoch 10/20
35/35 - 8s - loss: 0.0068 - 8s/epoch - 240ms/step
Epoch 11/20
35/35 - 8s - loss: 0.0062 - 8s/epoch - 241ms/step
Epoch 12/20
35/35 - 8s - loss: 0.0062 - 8s/epoch - 242ms/step
Epoch 13/20
35/35 - 9s - loss: 0.0060 - 9s/epoch - 244ms/step
Epoch 14/20
35/35 - 8s - loss: 0.0099 - 8s/epoch - 243ms/step
Epoch 15/20
35/35 - 8s - loss: 0.0062 - 8s/epoch - 235ms/step
Epoch 16/20
35/35 - 9s - loss: 0.0049 - 9s/epoch - 249ms/step
Epoch 17/20
35/35 - 9s - loss: 0.0052 - 9s/epoch - 246ms/step
Epoch 18/20
35/35 - 9s - loss: 0.0052 - 9s/epoch - 250ms/step
Epoch 19/20
35/35 - 9s - loss: 0.0060 - 9s/epoch - 251ms/step
Epoch 20/20
35/35 - 9s - loss: 0.0057 - 9s/epoch - 251ms/step
```

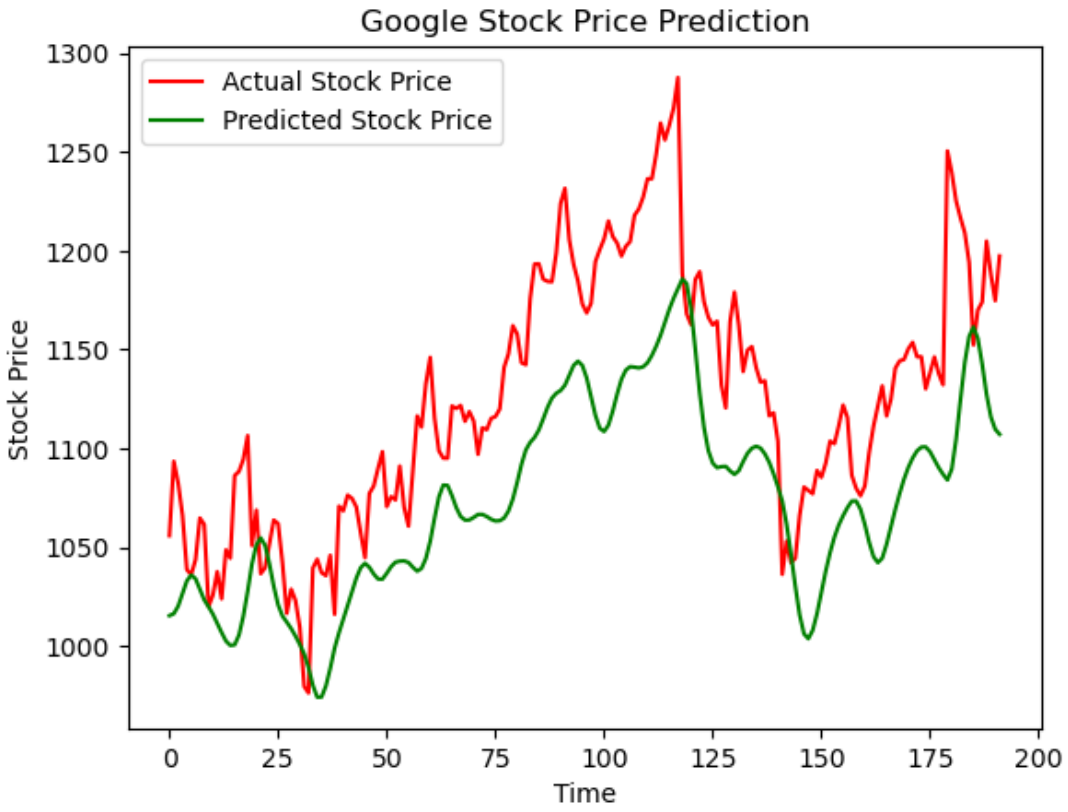
I also plotted the training model loss during the training process of the LSTM model. It visualizes the change in the model's training loss with each epoch of training.



Results and Discussion:

I used the trained LSTM model to predict the closing prices of Google for the testing data. I then transformed the predicted values back to their original scale using the inverse_transform method of the MinMaxScaler object. The predicted values were then plotted against the actual closing prices.

The plot shows that the model is able to capture the overall trend of the stock prices and predicts the turning points reasonably well. However, there are instances where the model fails to predict sharp peaks and dips in the prices. This could be due to various factors such as sudden global events, company announcements, etc., that the model may not have considered.



Conclusion:

In this project, I successfully built a LSTM model to predict the stock prices of Google. I achieved reasonable accuracy in my predictions and showed that deep learning models can be effective in predicting stock prices. However, it is important to note that stock prices are highly volatile and depend on numerous factors that are difficult to predict. Hence, investors should use caution and consider multiple sources of information before making investment decisions.