Stock Market Prediction for Apple and Amazon using LSTM

Imaan Shahid

Abstract

It is an established fact that the market is unstable and keeps changing with every second, hence a lot of research has been conducted in regard with accurately predicting the stock prices of different companies to aid in the investment decisions of the companies and the investors. Accurate prediction of stock market prices would allow the investors to gain significant profits. This project aims to predict stock market prices for companies with the highest stocks - 'Apple' and 'Amazon' – using LSTM models and achieving a low error value.

Keywords: Stock Market Prediction, LSTM, Apple, Amazon, Investment

Literature Review

Stock market prediction has been the subject for research for a long period of time and various methods using machine learning have been explored to get an accurate prediction of the stock market. Among all the machine learning techniques that have been used, the one proven to be the most effective has been Long – Term Short Memory Model (LSTM). LSTM is a type of Recurrent Neural Network, which works on providing feedback by saving the output from a particular layer and providing it to the input layer for decision making. LSTM works on the above-mentioned technique and is efficient in working on time dependent data, by processing entire sequences of data instead of a single point (like in CNN). One study found that an LSTM model outperformed both ARIMA and traditional neural networks in predicting stock prices. The study used historical stock price data from the NSE (National Stock Exchange of India) and found that the LSTM model had a lower mean absolute error and mean squared error than the other methods. LSTMs have been used in a wide range of applications such as natural language processing, speech recognition, and time series forecasting. They have been found to be particularly effective at handling sequential data, where traditional RNNs struggle, and have been shown to have the ability to handle long-term dependencies in the data.

Technical Difficulty

The uniqueness of the LSTM model is that it has a memory for the recent sequences that it encounters which enables it to process the data with long term dependencies. An LSTM cell contains various components important for it to accurately predict stock prices namely: input gate, forget gate, and an output gate. The gates control the information that is to be passed or discarded. The input gate is responsible for controlling the flow of new information to the model by using a sigmoid function for deciding what parts are to be passed as the input. Furthermore, the forget gate controls the flow of information out of the cell and decides which parts of the past state must be discarded and what must be used. Lastly, the output gate has the sole function of deciding whether the current state is to be passed as the output of the layer, the control of this flow is also carried out by using a sigmoid activation function. The sigmoid function used in the LSTM model is the Tanh function whereby the negative outputs are labelled strongly negative, and the zero values are mapped near zero in the tan hyperbolic graph.

In addition to this, one of the main principles used by the LSTM model is back propagation through time which calculates new/updated weights to adjust the weights of the gates and the cell state to reduce the error between the predicted data and the actual data. Hence the complete training of the model involves processing the time-series data (stock market dataset) as a whole sequence whereby the mentioned gates decide which outputs and inputs must be forwarded to the next layers to get an accurate predicted value. Along with this, the weights of the model are continuously updated according to the data to ensure lowest error.

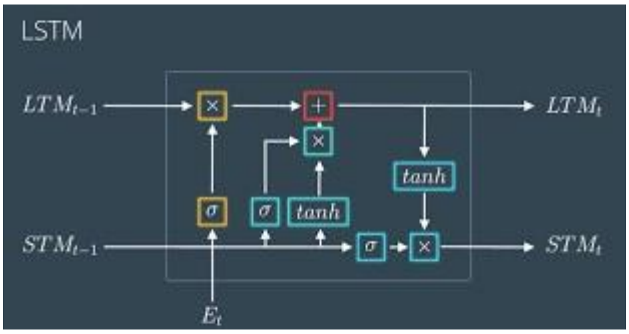


Fig 1: LSTM Architecture

Experimental Methodology

As explained above, in this project two LSTM models were used to accurately predict the stock market prices from Apple and Amazon. Following, we describe the processes and methodology we used to develop this project.

Dataset Acquisition

The first step carried out for this purpose was getting the previous data of the closing prices of stock for both the companies. Using the Yahoo Finance API known as “yfinance”, we extracted the dataset for both the companies for the last 5 years and plotted them to analyse and view the trends.

Data Preprocessing

The first step of data preprocessing was data cleaning and creating a training dataset from the entire data that we extracted. This was done by extracting only the closing stock price column from the dataset and then splitting the resulting data into 80% training data for the LSTM model and 20% testing data to test on unknown data.

In the next step the stock price data was scaled between 0 and 1, along with creating two NumPy arrays the first one containing the stock prices for the last 60 days (e.g., 0 to 59), and the second one containing the value to be predicted by the model (e.g., 60th value). Lastly the data was reshaped to a three-dimensional array as the LSTM model only accepts inputs with a three-dimensional shape.

Model Training

We used two different LSTM architecture for predicting the stock prices for the companies depending on the lowest error. The first model architecture included 4 layers of the LSTM model with 100 units of output space. The first 3 layers returned the sequences as feedback while the last layer did not as it was the output layer. Drop out layers were used with each layer with a rate of 0.5 to prevent overfitting of the data. In the end a Dense (Fully Connected) layer was added to take output from all neurons and create an output for the model. The second architecture contained two layers of the LSTM model and two dense layers along with dropout layers.

The model was trained with a batch size of 32 and epoch size of 100, by fitting it on the x and y train datasets and evaluating the model based on the root mean square error.

Model Evaluation

In the last step of the project, a testing dataset was created the same way as the training dataset, after which the prediction of the stock market price was made by the model on the testing data. The prediction was converted to its original form by performing an inverse scaling. To calculate the root, mean square error, the difference between the prediction and actual value was calculated and input into the RMSE formula.

Analysis and Conclusion

First the data for the stock market was visualized to notice the trends in the stock market and accordingly train the model for training. The following figures are the trends of stock market prices for Apple and Amazon.

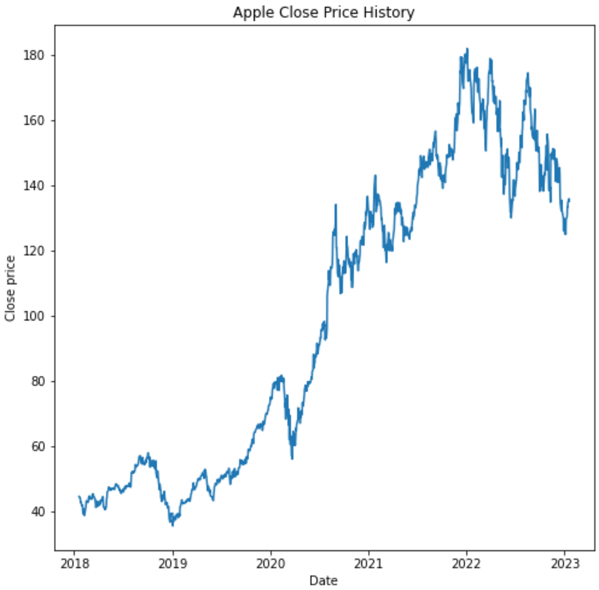


Fig 2: Apple Close Price History

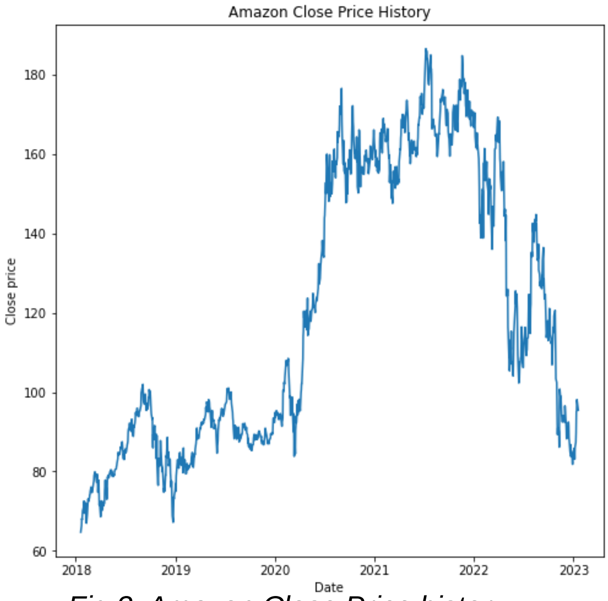
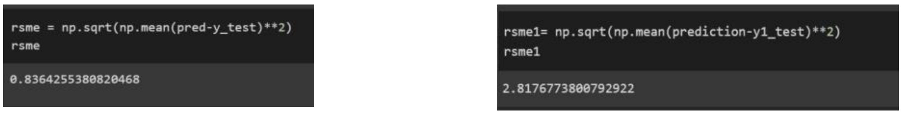


Fig 3: Amazon Close Price history

The model was evaluated by the Root Mean Square error, and we were able to achieve accurate results by getting a value of approximately 0.8364 for the first model and 2.817 for the second model, which are relatively accurate results.



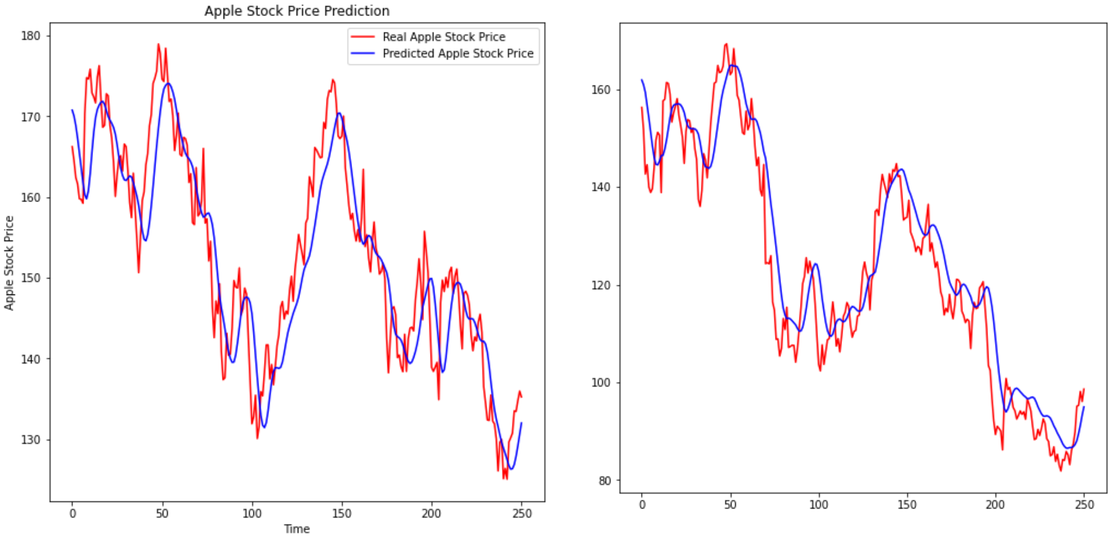
```
rsme = np.sqrt(np.mean(pred-y_test)**2)
rsme
0.8364255380820468
```

```
rsme1= np.sqrt(np.mean(prediction-y1_test)**2)
rsme1
2.8176773800792922
```

Fig 4,5: RSME values for both models

At last, the actual prices and predicted prices were compared by plotting them against the time axes, which can be seen by the following figures.

Fig 5,6: Actual and Predicted graphs for Apple and Amazon


Apple Stock Price Prediction
Real Apple Stock Price
Predicted Apple Stock Price
Apple Stock Price
Time