Course Project

Moon, Hyosik

September 10, 2021

1 Required

- **Main objective of the analysis**
  I will analyze the Goldman Sachs's Stock Data from 1999-05-04 to 2021-07-01 with time series models such as RNN and LSTM.

- **Brief description of the data set**
  In the data, there are seven features such as 'Date, Open, High, Low, Close, Adj Close, and Volume' (Figure 1). We will use 'Date' and 'Close' features to predict stock price. To be specific, we will train models with for about past 20 years data and predict the upcoming 7 months' Goldman Sachs's stock price.

	Open	High	Low	Close	Adj Close	Volume
count	5577.000000	5577.000000	5577.000000	5577.000000	5577.000000	5.577000e+03
mean	153.388107	155.255690	151.518473	153.413700	134.953673	5.347931e+06
std	59.548650	59.940215	59.125804	59.536651	60.874014	6.154837e+06
min	54.000000	54.540001	47.410000	52.000000	43.174923	1.076000e+05
25%	99.375000	101.070000	97.809998	99.300003	81.400467	2.525500e+06
50%	155.300003	157.000000	153.720001	155.330002	134.594437	3.598300e+06
75%	194.839996	197.399994	192.449997	194.970001	173.237244	5.660900e+06
max	393.000000	393.260010	387.549988	391.450012	391.450012	1.145907e+08

Figure 1: Data description without date

- **Brief summary of data exploration** I remained only 'Data', and 'Close' features. In order to use for RNNs and LSTMs it needs a 3D numpy array of shape (n_sample, time_steps, n_features). As I learned in the class I made a function to preprocess train_test data sets. And I train the RNN models with 'cell_units = 30, epochs = 100, loss=mean_squared_error, optimizer=adam'.

- **Summary of training at least three models** I implemented the data with 'RNN, LSTM' models.

    1. Recurren Neural Networks. I trained RNNs with 5577 days' data and predicted following 420 days. As you expected the result was very poor. It was an obvious result becuase RNN's prediction accuracy is very bad for long-term period. (Figure 2)

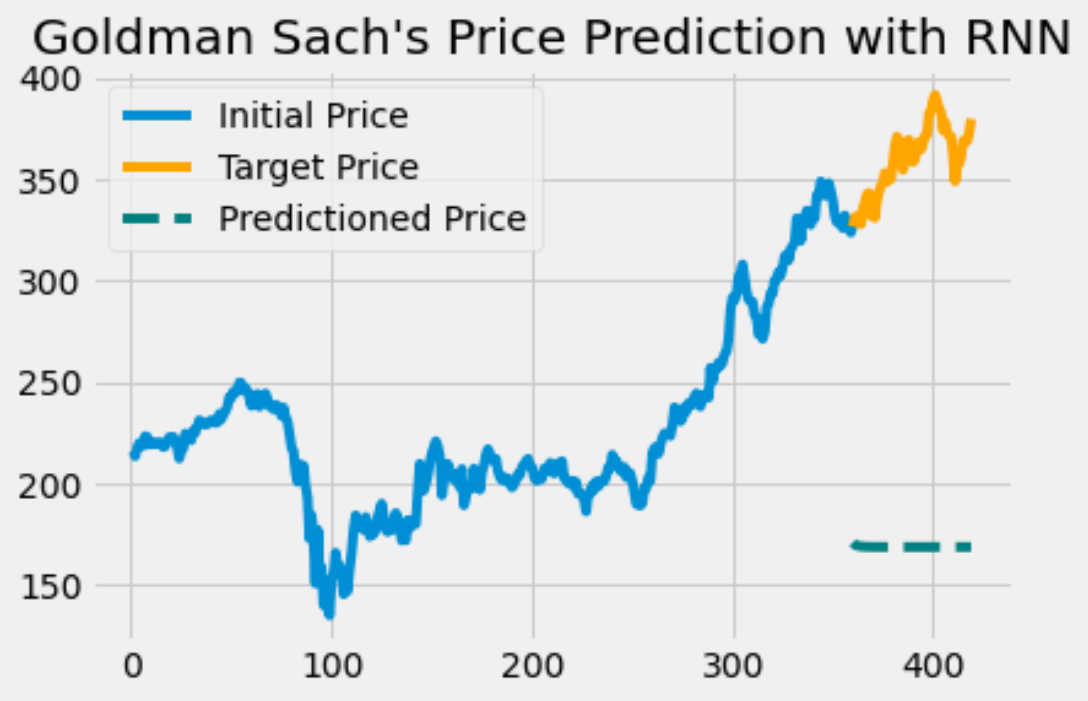

Figure 2: Recurrent Neural Networks

    2. LSTM. Unfortunately the result was also bad (Figure 3).

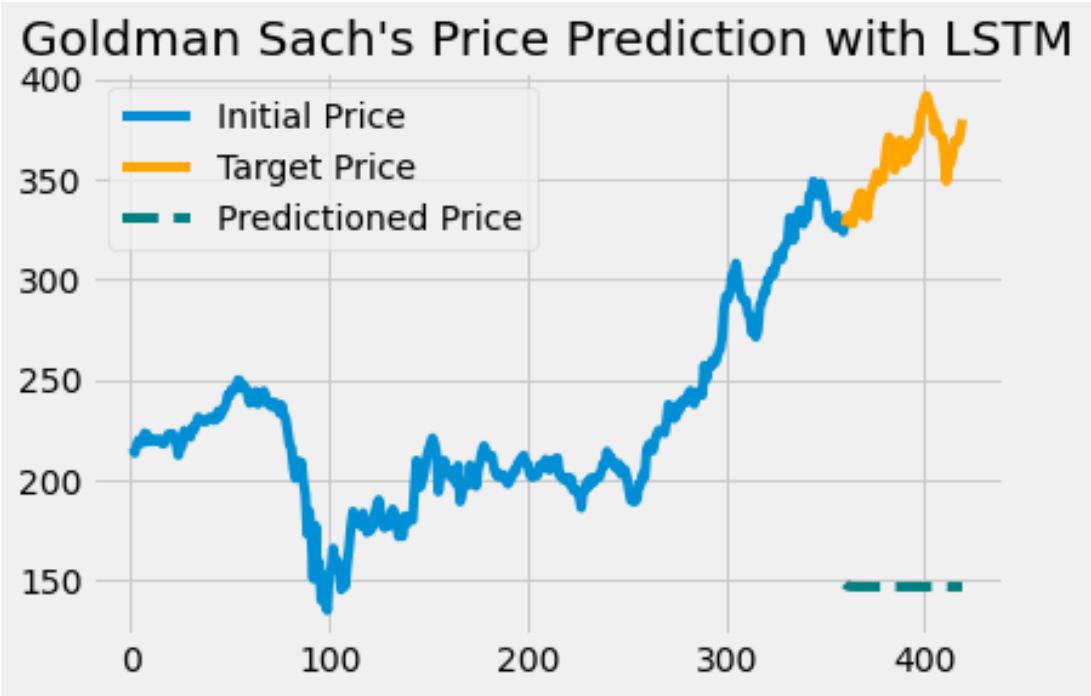

Figure 3: Long Short Term Memory

- **Explanation of your final model** I needed to analyze the result. When I inspected the codes, it seemed not to have an error. Thus, I modified the length of data from (5577, 360, 1) to (1260, 210, 1) and epochs from 300 to 1200. Interestingly RNNs also showed much better result even I trained the model with fewer data (Figure 4). However, for LSTM it showed still bad predction accuracy (Figure 5). So, I trained more but with short date, i.e. from [data = (1260, 210, 1), nodes = 30, epochs = 1200] to [data = (1260, 210, 1), nodes = 70, epochs=1400] (Figure 6). Because it still shows bad result, I added more nodes to 100 and trained again (Figure 7). It looks a little bit better but still quite bad. When I see the trained result the prediction error was low, so I thought that the reason of low accuracy maybe is for the number of nodes not the epochs. So I inceased the number of nodes from 100 to 210 which is same number of the time_steps and decrease the epochs from 1200 to 1000 to reduce the training time. It shows still bad result.. So I changed the length of training data from 1260 to 2000 and decrease the epochs to 350 (Figure 9). As the result of 7 analysis, all models seem not to be appropriate to predict the stock price. However the last one (LSTM6) shows a liitle bit better predction result compared to the others.

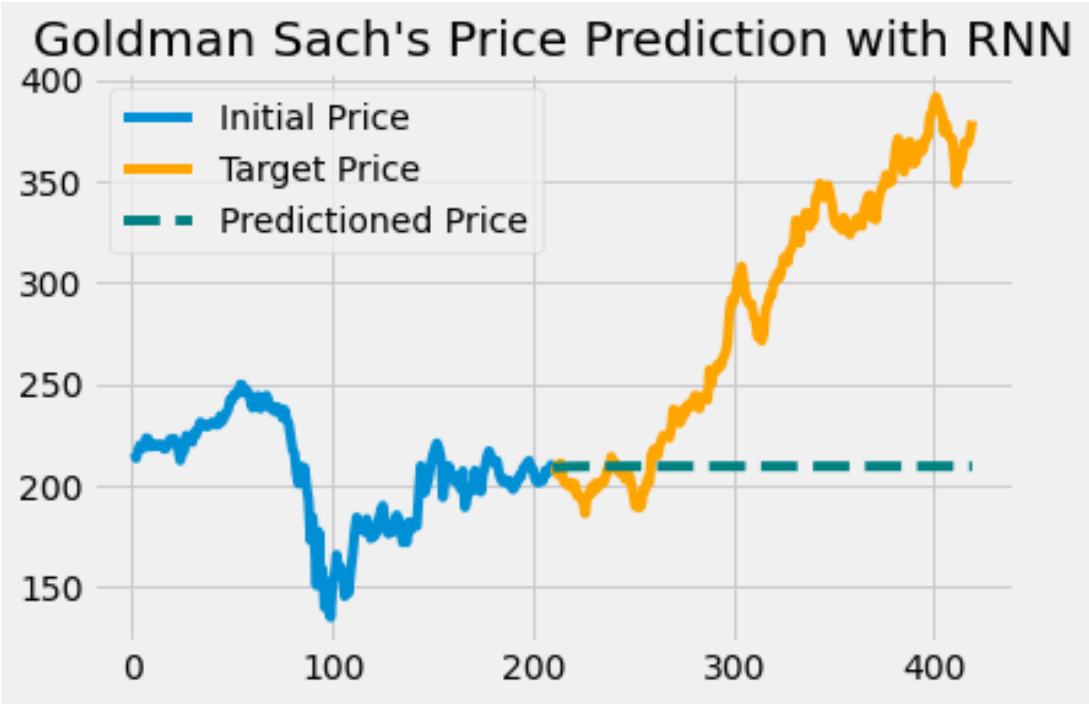

Figure 4: RNN with data = (630, 210, 1), nodes = 30, epochs = 1200

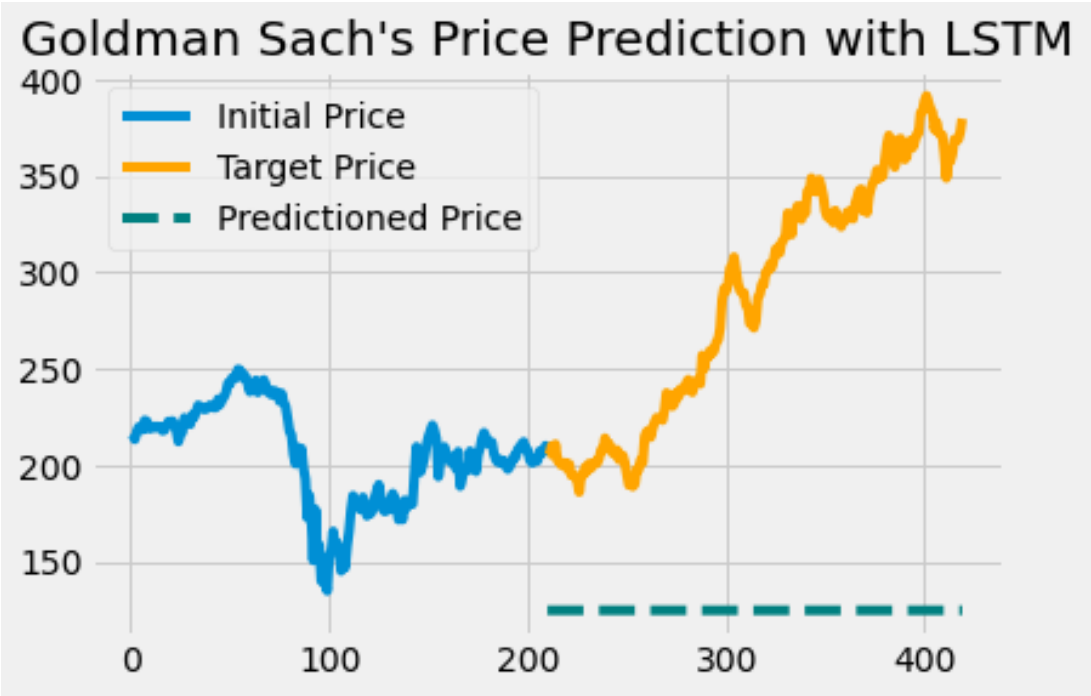

Figure 5: Long Short Term Memory2 = (1260, 210, 1), nodes = 30, epochs = 1200

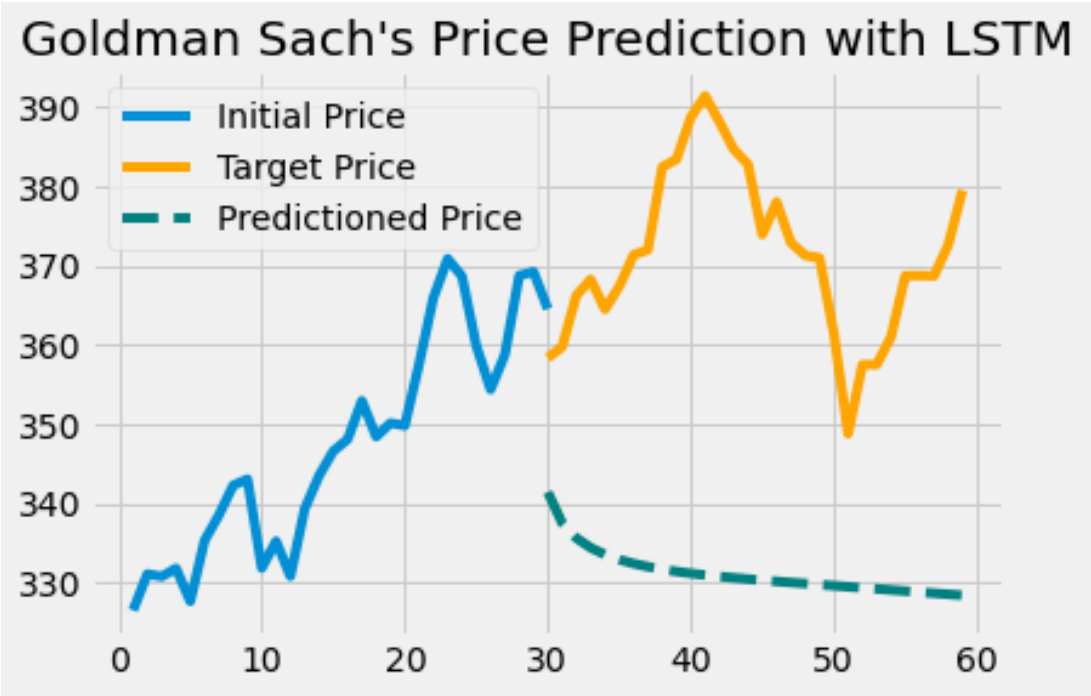

Figure 6: Long Short Term Memory3 = (1260, 210, 1), nodes = 70, epochs = 1400

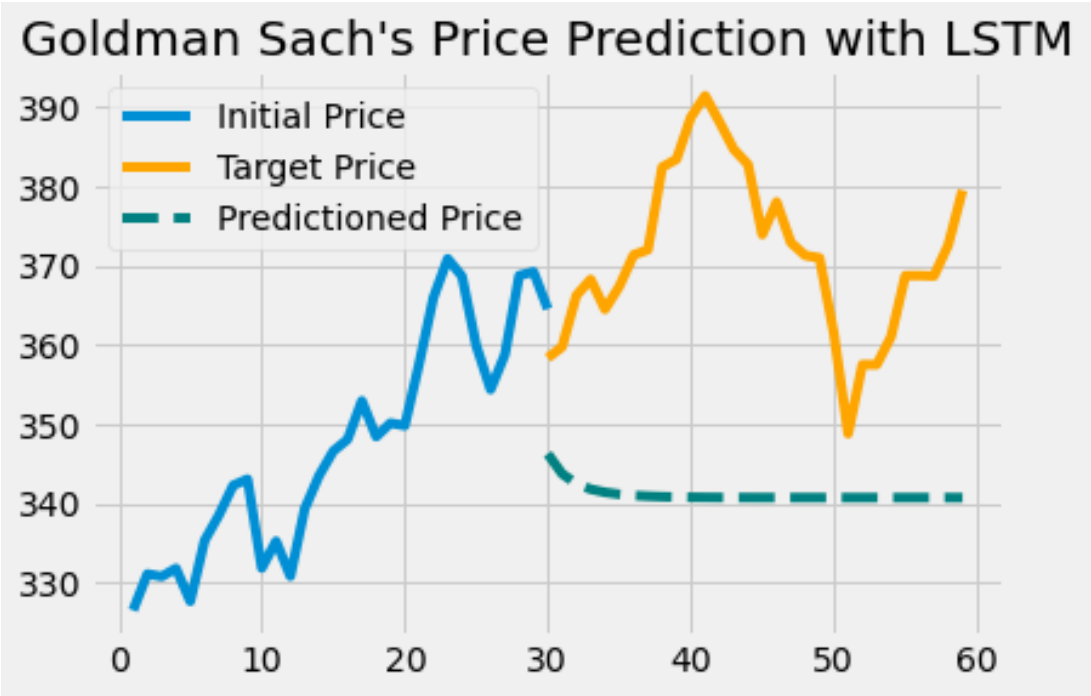

Figure 7: Long Short Term Memory4 = (1260, 210, 1), nodes = 100, epochs = 1200

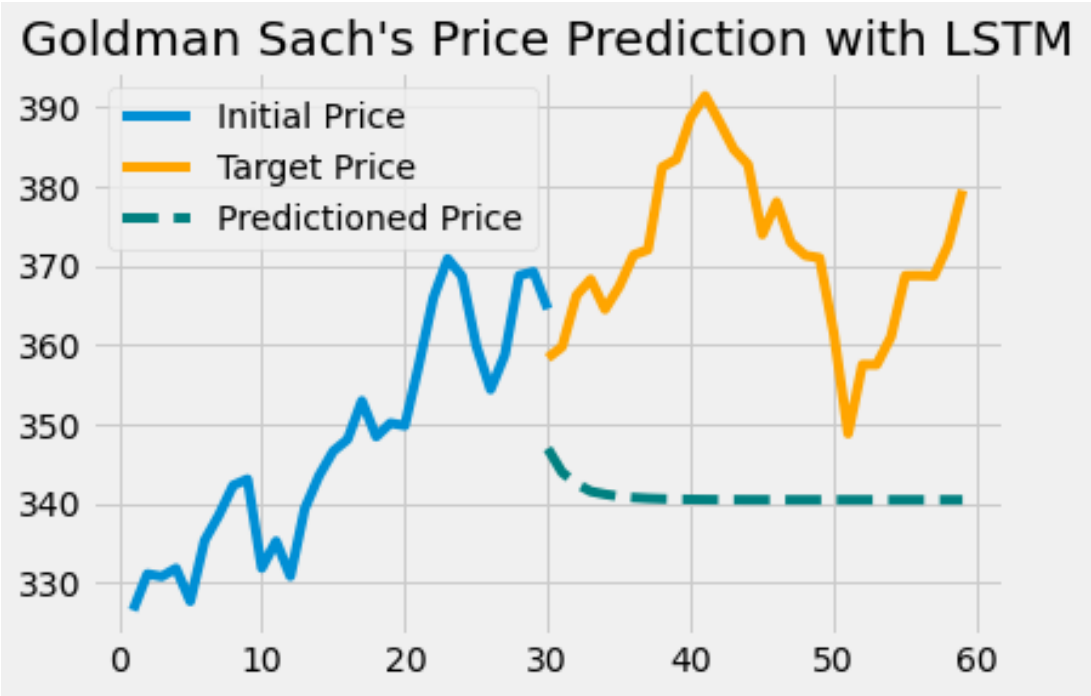

Figure 8: Long Short Term Memory5 = (1260, 210, 1), nodes = 210, epochs = 1000

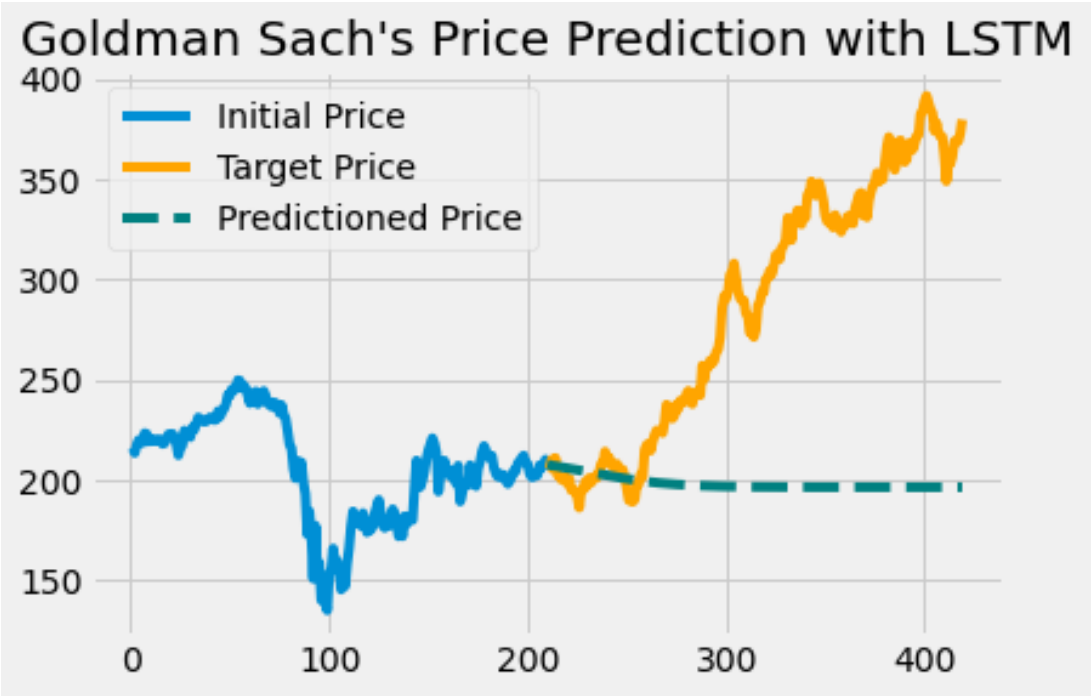

Figure 9: Long Short Term Memory6 = (2000, 210, 1), nodes = 210, epochs = 350

- **Summary Key Findings and Insights** In the real world, most of data are non-stationary data. So, we need to manipulate the data to stationary data by subtracting trends and seasonality. Then the data should show constant mean and variance, which is a very picky condition. Thus, it needs expertise to analyze and manipulate the data correctly. However, in the case of Deep Learning Models for time series analysis such as RNNs and LSTM, doesn't need to manipulate data to make stationary. It's really incredible fact because even if not an expert can predict the time series data if they have a little knowledge and skills about deep learning. I think it's the magic of the deep learning. But in order to gurantee the accuracy of the model and interpretation of the models we need deep theoroetical knowledge, high comuputing power, and huge amount of data. If not, as we've seen in this report, we can't gurantee the reliability of the model and the result.

- **Suggestions for next steps** In oder to predict the stock price more accurately, I think using the ground truth values seems to be more appropriate rather than predicting following values with previously predicted values (Figure **??**). In the figure (LSTM7) we can see that after 300 days the model seems to stop to predict. There seems to be an error in the model.. We need to analyze the reason of it. And

I was sorry that I wasn't able to use Facebook Prophet models to predict the time series data. It had a problem to install, and I wasn't able to figure it out. If I can solve the installing issure, I'd like to use Facebook Prophet libraries for the time series analysis.

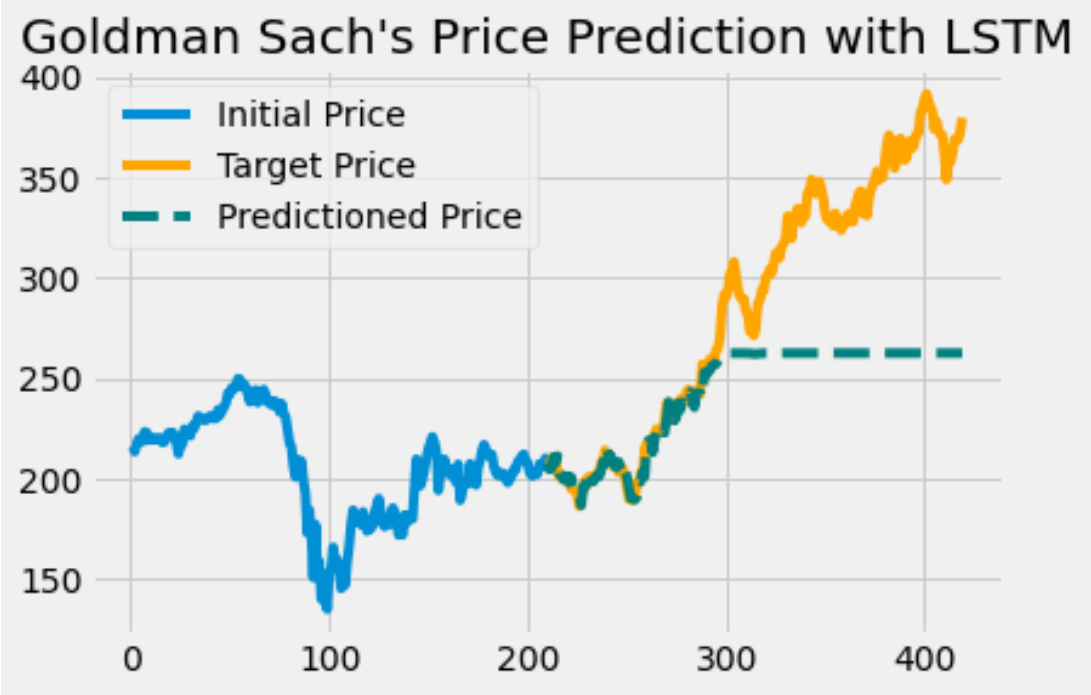

Figure 10: Long Short Term Memory7 = (2000, 210, 1), nodes = 210, epochs = 350