Predicting Stock Prices

Tanjin Sharma
Ronica Gupta

Table of Contents

01

The Idea

Why this project excited us!

PROBLEM

The stock market is exciting and a great way to invest and make money. But how to choose which stocks to invest in, without any financial knowledge?

SOLUTION

As data Scientists, we can leverage available data to create a prediction model that helps us understand the performance of stocks in the market and then invest money accordingly

02

The Data

What data did we use?

The Data

- We extracted the data from **https://finance.yahoo.com/** which hosts the stock prices till date for the companies listed on the stock market.
- We used the **pandas_datareader** to get data from Yahoo Finance.
- We took Google stock data from 2010- 2022(April).
- The API returns the Closing price, Start price and the volume of stocks, traded for each day.

03

The Modelling

How did Deep Learning Help?

How does Deep Learning Help

- For the modelling of this data, we used the LSTM model.
- LSTM layers have an inherent property to control the inflow and outflow of only the relevant information from the past combined with the present information.
- This is vital in the context of stock price prediction as they largely depend on the historical data.

How does Deep Learning Help

```
# Defining the model
# This model uses LSTM layers

class LSTM(nn.Module):
    def __init__(self, input_dim, hidden_dim, num_layers, output_dim):
        super(LSTM, self).__init__()
        self.hidden_dim = hidden_dim
        self.num_layers = num_layers
        self.lstm = nn.LSTM(input_dim, hidden_dim,
                            num_layers, batch_first=True)
        self.fc = nn.Linear(hidden_dim, output_dim)

    def forward(self, x):
        h0 = torch.zeros(self.num_layers, x.size(
            0), self.hidden_dim).requires_grad_()
        c0 = torch.zeros(self.num_layers, x.size(
            0), self.hidden_dim).requires_grad_()
        out, (hn, cn) = self.lstm(x, (h0.detach(), c0.detach()))
        out = self.fc(out[:, -1, :])
        return out
```

04

The Process

What went well and what issues did we face?

What went well

- The process of downloading the data was very smooth and fast.

- The training was also very fast on the CPU.

What issues did we have

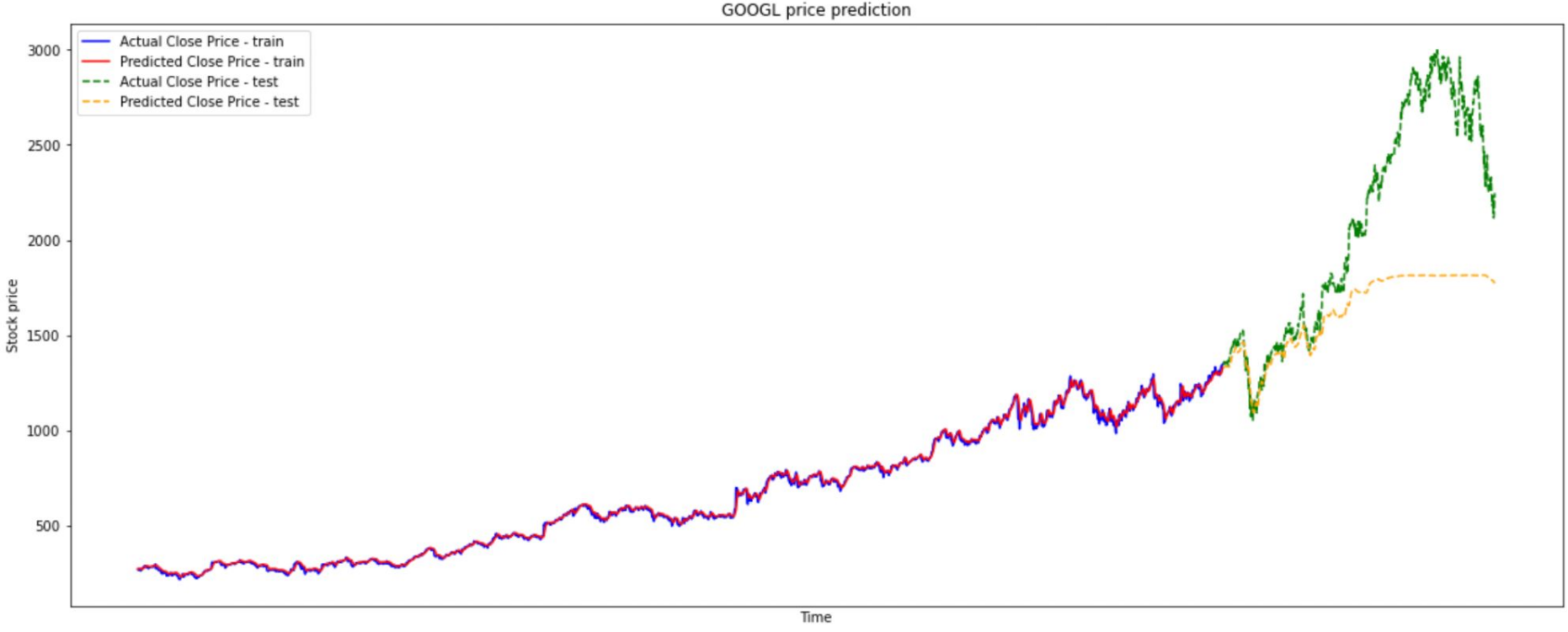
- A lot of variation and fluctuation in data causes poor predictions in long term
- Predictions flat-line because of the same
- The need to train the model frequently to incorporate the historical effects with the new data

05

The Conclusion

How is the final result?

Results and Conclusion



Results and Conclusion

- Number of Epochs run : 800
- Training time : 300s
- Mean Squared Error (train) : 0.000139
- Mean Squared Error (test) : 0.185

Conclusion and Future Goals

- The predictions are not much accurate for the future data
- A more complex model might be helpful that can capture the trend and fluctuations
- A good approach might be to consider the extrinsic factors of the market as well.

Thank You