Stock Price Prediction

- $SPY (S&P 500 ETF)
- $BAC (Bank of America)
- $WMT (Walmart Inc.)

Edward Lee

CISC 3440
Fall 2022

Professor
Matthew McNeill

Table of Contents

Colab Notebook

Notebook: Colab_Notebook

Introduction

Predicting the stock market's price is the essential for many traders and investors. With slight changes in a penny can cause thousands of dollars if not millions of dollars. Currently given the global economic situations, stock prices are at the downward trend or volatile at the moment. Many investors lose thousands and millions of dollars every day. Therefore, I came up with a question that what if the machine learning can predict or estimate the opening price of a particular stock, and the investors can trade safely and have the estimate value in return when they trade. I asked, "What is the opening price of $SPY (S&P 500 ETF), $BAC (Bank of America), and $WMT (Walmart Inc.)?"

Method

The dataset

I acquired the dataset from Kaggle called "Huge Stock Market Dataset", it contains stocks and ETFs dated back to 1972 to 2017 (some datasets are from 2005 to 2017). The dataset contains Dates, Open (stock price when the market open), High (the highest peak during the day), Low (the lowest point during the day), Close (the price when the market closes), Volume, and OpenInt. I omitted the Volume and OpenInt columns as they will not be used in the predictions.

What data cleaning and preprocessing steps did you perform?

Firstly, I downloaded the dataset text files and converted them to CSV file, then uploaded them on to GitHub, so that I can access the dataset easily with GitHub raw CSV file inside the Google Colab. Inside the Colab, import the datasets, identify the number of rows and columns for each datasets. Lastly, split the dataset into a training set and a testing set.

What machine learning methods are you using?

I have decided to use Simple Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) methods to predict the price of the opening. I used Simple RNN and LSTM because the Simple RNN is a basic type of RNN that does not have any gating mechanism and LSTM can solve the vanishing gradient problems. During the process of making the models for simple RNN, I used hyperbolic tangent for the activation functions, Mean Square Error function, and Adam optimizer (a combination of momentum optimizations and RMSProp). The steps I took for the predictions and comparison between RNN and LSTM:

1. Data Preprocessing
2. Normalizing the Data
3. Making the Model with Simple RNN
4. Prediction with Simple RNN

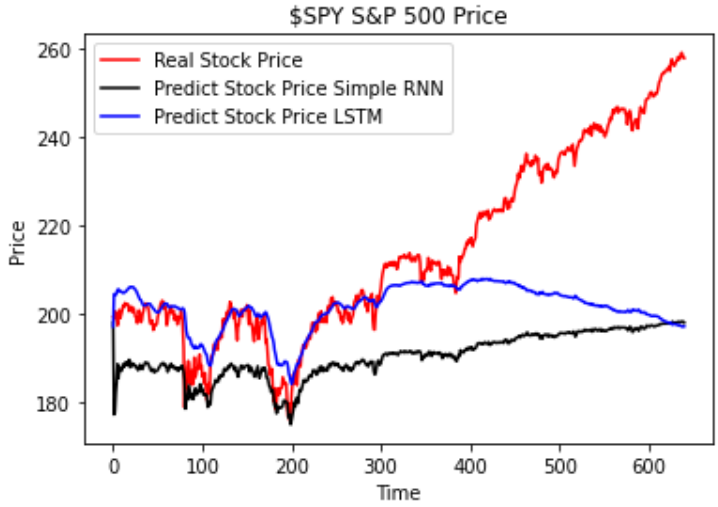
5. Evaluating the Model
6. Making the Model with LSTM
7. Prediction with LSTM
8. Comparison with Real Data, Simple RNN, and LSTM

Experiments

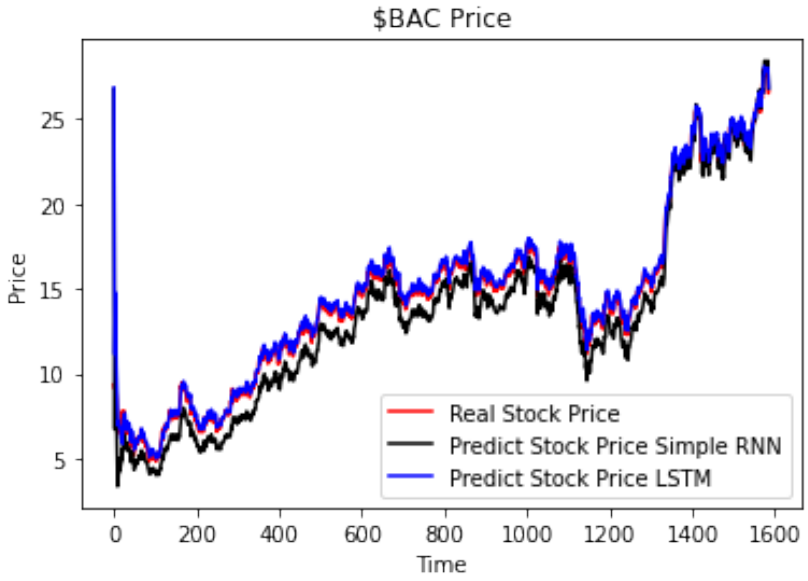
I used three different stocks and three different sizes of the dataset. $SPY has the total of 3,201 rows, $BAC has the total of 7,929 rows, and $WMT has the total of 11,443 rows. I experimented with different values for timesteps in the prediction stage and different values for batch_size in modelling stage. Some displayed the expected results and some displayed unexpected and disappointing results.

Results and discussion

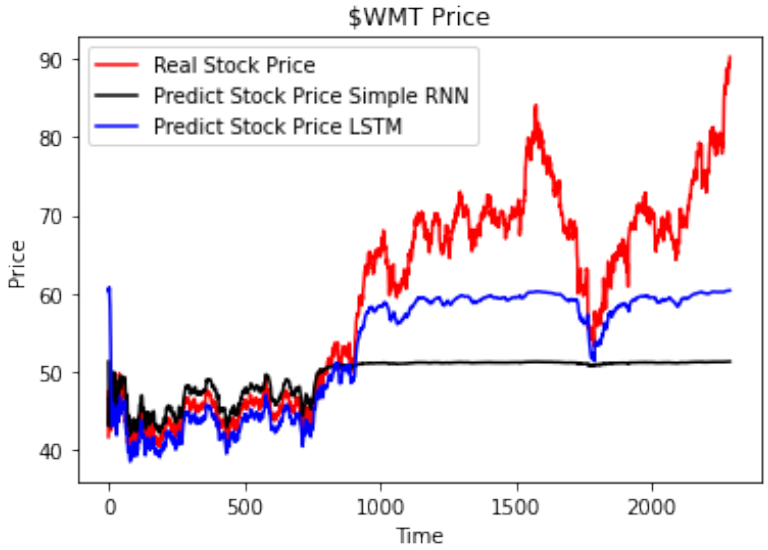
During the experimentation with the datasets, I have encountered many suspicions and errors. I found out that $SPY does not have enough data to train and predicts, there are total of 3,201 rows in $SPY dataset. After the split, there are 2,560 rows for the training set and 641 rows for the testing set. However, both RNN and LSTM displayed two unique different results, which makes me think, it is possible that there are not enough training dataset therefore, it caused a disappointing result. I changed the value of the batch_size and value of the timesteps, but the prediction is nowhere close to the real price.



However, for $BAC the real price, Simple RNN, and LSTM are very close to each other. The dataset for $BAC is larger than $SPY and it has the total of 7,929 rows of data. After the split, there are 6,343 rows of training set and 1,586 rows of testing set. Therefore, I assumed that since there are more amount of training set and testing set for $BAC it is most likely that machine can learn more about the price movement.



To prove my hypothesis is correct, I needed to test with another stock that has larger dataset. I chose $WMT to test my hypothesis and it displayed a discombobulated result. $WMT has total of 11,443 rows of data, after the split, it has 9,154 and 2,289 respectively for the training set and the testing set. At the beginning of both predictions, they practically near the real price, however, approximately around 1000 seconds point both Simple RNN and LSTM output very differently.



Conclusion

From this project, I have learned that the investor should not really depend on the machine learning for the stock market price. And the machine learning needs more data to get trained and tested extensively to get an accurate results.

Citations

Biswal, Avijeet. “Stock Price Prediction Using Machine Learning: An Easy Guide: Simplilearn.” *Simplilearn.com*, Simplilearn, 15 Nov. 2022, https://www.simplilearn.com/tutorials/machine-learning-tutorial/stock-price-prediction-using-machine-learning.

Marjanovic, Boris. “Huge Stock Market Dataset.” *Kaggle*, 16 Nov. 2017, https://www.kaggle.com/datasets/borismarjanovic/price-volume-data-for-all-us-stocks-etfs.