Time Series: Stock Return Prediction

Zixin Huang

July 29, 2021

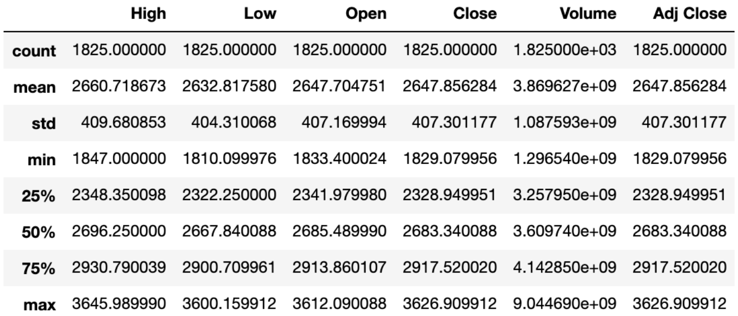
1 Data and Objective

The goal of this project is to predict the stock data and help investors make decisions with regards to their portfolio. We do so by assess the ability of prediction of different time series models, including ARMA, RNN and LSTM.

The data[1] contains about 5 years of the price information of S&P500 from 2015-11-23 to 2020-11-20, having 1825 observations and 7 columns. The columns indicate the date, highest price, lowest price, open price, close price, adjusted close price, and trade volume. There is no missing value in this dataset. The statistical summary of the data can be seen in Figure 1. The summary shows that the price of S&P500 ranges from $1800 to $3650, and the daily trade volume is very high.

One of the possible difficulties for this analysis is that we cannot ensure the data is stationary, which is the key assumption in econometric analysis. Another problem may be the limited number of observations in this dataset.

Figure 1: Summary of Data



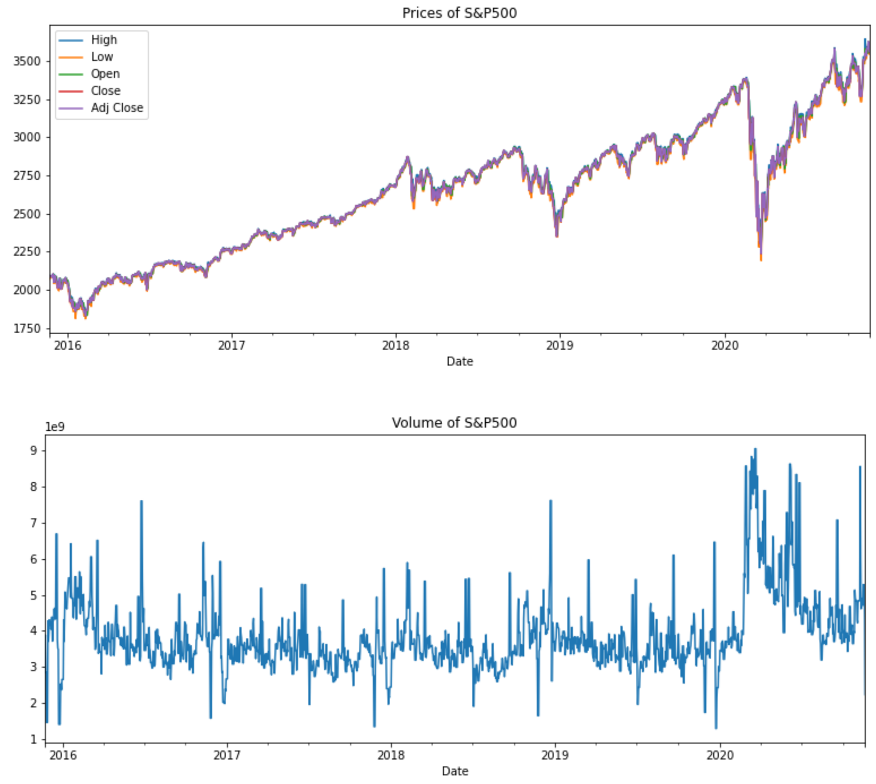
	High	Low	Open	Close	Volume	Adj Close
count	1825.000000	1825.000000	1825.000000	1825.000000	1.825000e+03	1825.000000
mean	2660.718673	2632.817580	2647.704751	2647.856284	3.869627e+09	2647.856284
std	409.680853	404.310068	407.169994	407.301177	1.087593e+09	407.301177
min	1847.000000	1810.099976	1833.400024	1829.079956	1.296540e+09	1829.079956
25%	2348.350098	2322.250000	2341.979980	2328.949951	3.257950e+09	2328.949951
50%	2696.250000	2667.840088	2685.489990	2683.340088	3.609740e+09	2683.340088
75%	2930.790039	2900.709961	2913.860107	2917.520020	4.142850e+09	2917.520020
max	3645.989990	3600.159912	3612.090088	3626.909912	9.044690e+09	3626.909912

2 EDA and Feature Engineering

Figure 2 gives a visualization of the dataset. We can see that, in general, the price have a trend that steadily going up, and the trade volume remains between 2×10^9 and 6×10^9. There were times, however, the price rapidly decreases such as in the beginning of 2019 and March 2020. In particular, the price drop in March 2020, which was due to the COVID-19, was accompanied with significantly higher trade volume, reflecting that the public was panic and lost faith in the economy. In the following months, the trade volume remained high and the price was gradually going back up, suggesting the effect of the shock was fading and the public expected the economy to be back to normal soon.

[1] https://www.kaggle.com/arashnic/time-series-forecasting-with-yahoo-stock-price

Figure 2: Visualization of Price and Volume

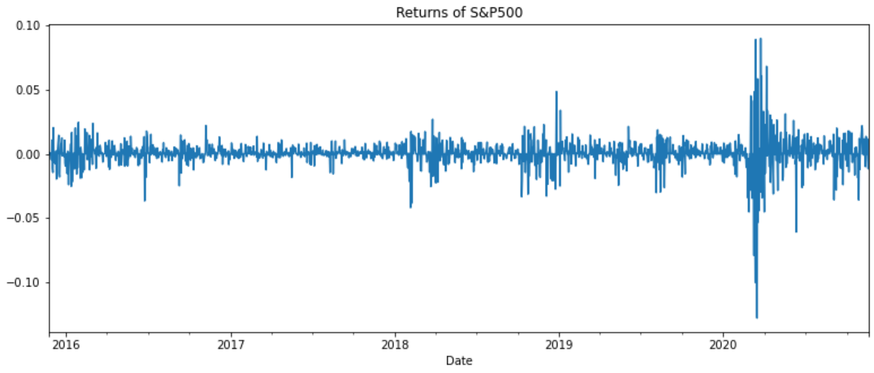


The price of the S&P500 index is clearly not stationary, since it has a trend and is possibly heteroskedastic due to the drop in 2020. Therefore, we analyze the log returns, which can be computed by

$$r_t = \log(p_t) - \log(p_{t-1}) = \log(\frac{p_t}{p_{t-1}})$$

where r_t stands for the log return at time t and p_t stands for the price at time t. Figure 3 shows the log returns of the stock index.

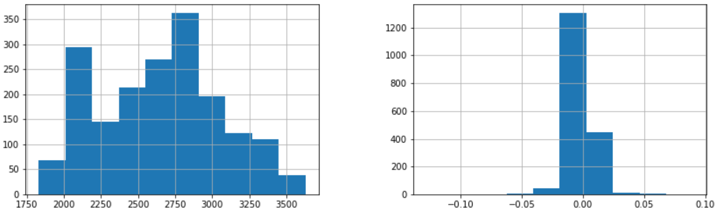
Figure 3: Log Returns of S&P500



We can see that after difference the price the trend was removed, and the returns center around 0.

However, although we use log transformation to damp the variance, the problem of heteroskedasticity may still remain. To further test whether the data is stationary, we use augmented Dickey–Fuller (ADF) test. The ADF test checks whether we can reject the null hypothesis that the data has a unit root (i.e. not stationary). The test statistics we have is -8.4389, which is smaller than the 1% critical value -3.4340, indicating that we are 99% confident to reject the null and assume that our data is stationary. Figure 4 shows the histograms of price and log returns.

Figure 4: Distributions of Price (Left) and Log Returns (Right)



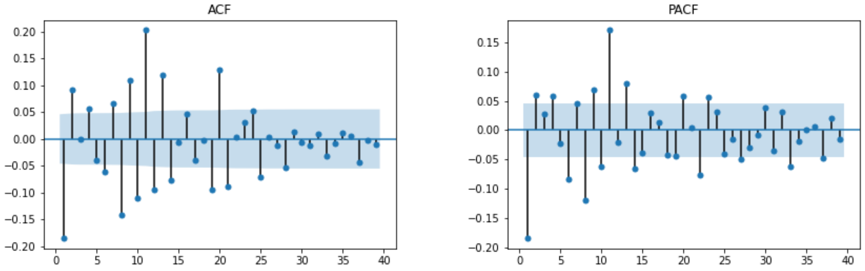
Before we build the models, we split our dataset into training and test set. We use the data before 2020 as training set and data after 2020 as test set, to assess our models abilities of prediction as well as abilities to dealing with "black swan" event.

3 Models

3.1 ARMA

To determine the orders of the ARMA model, we plot the auto-correlation function (ACF) and partial auto-correlation function (PACF) of the dataset (Figure 5).

Figure 5: ACF and PACF Plot



We can see that the auto-correlation decaying towards 0 as the number of lag increases. Lag 3 is the first number of lag that is insignificant both in ACF and PACF, thus we choose ARMA(2, 2) model, which can be represented as

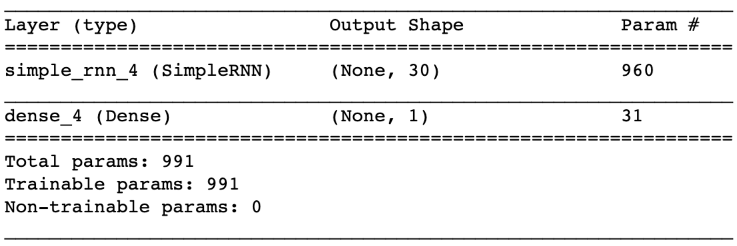
$$Y_t = \omega_0 + \omega_1 Y_{t-1} + \omega_2 Y_{t-2} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2}$$

where Y_t is the return at time t and ε_t is the regression error at time t.

3.2 RNN

The second model we choose is simple RNN, with only 1 hidden layer and 30 cell units. The summary of the model is shown in Figure 6. To train this model, we use ”rolling window” in our training set. For each window, the last value is the validation value used to compare with prediction based on all the previous values in the window. After predicting and comparing within a window, we move the window one step forward and repeat the process.

Figure 6: RNN Model Summary

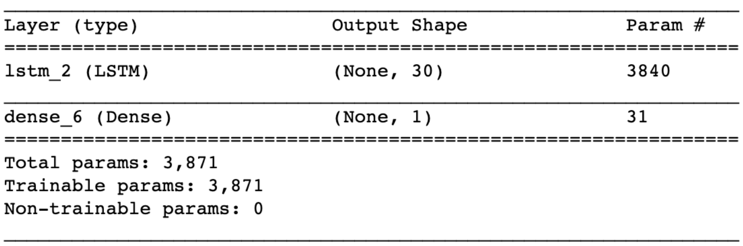


```
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
simple_rnn_4 (SimpleRNN)     (None, 30)                960       
_________________________________________________________________
dense_4 (Dense)              (None, 1)                 31        
=================================================================
Total params: 991
Trainable params: 991
Non-trainable params: 0
_________________________________________________________________
```

3.3 LSTM

The last model we choose is the LSTM model, with 1 hidden layer and 30 cell units. Figure 7 shows the summary of the model. The training process of the LSTM model is similar to the process of the RNN model. From the summary we can see that there are more parameters in the LSTM model compare to the RNN model, therefore the LSTM model may have more risk of overfitting.

Figure 7: RNN Model Summary

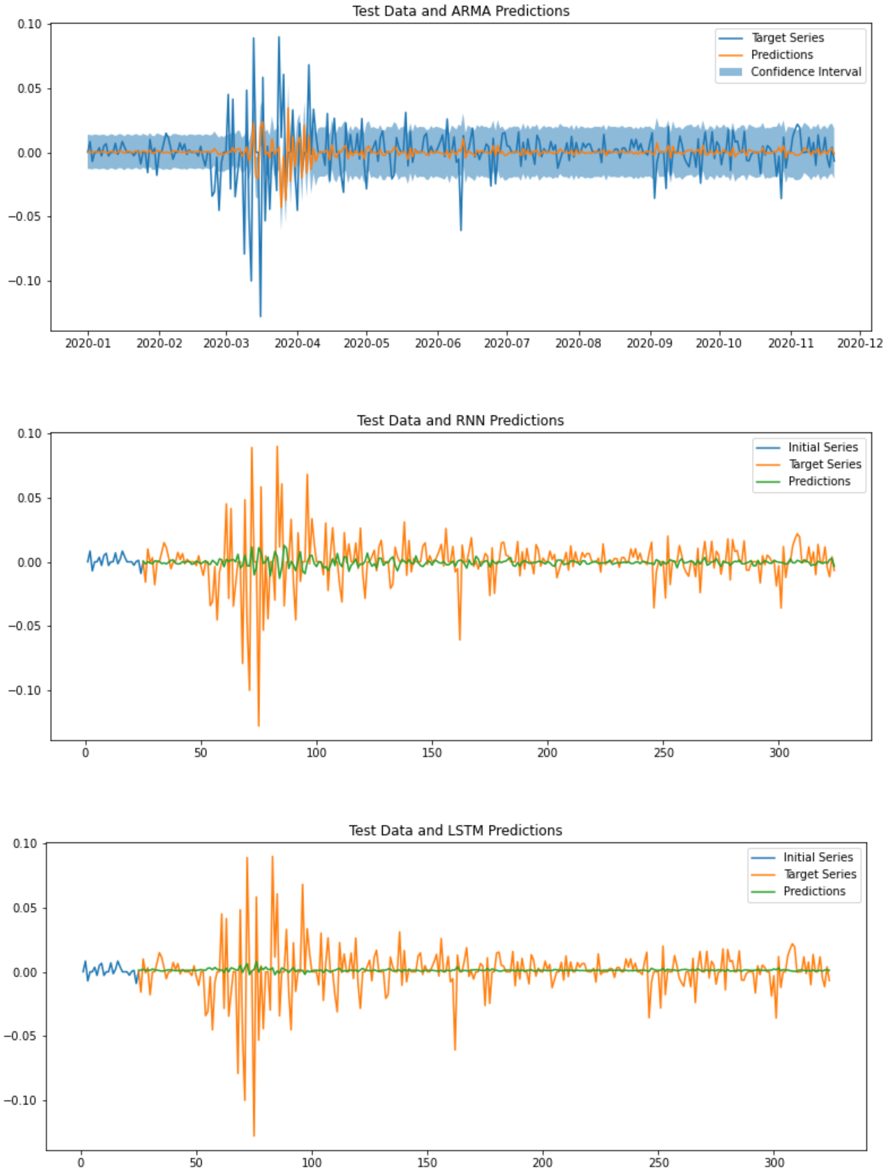


```
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
lstm_2 (LSTM)                (None, 30)                3840      
_________________________________________________________________
dense_6 (Dense)              (None, 1)                 31        
=================================================================
Total params: 3,871
Trainable params: 3,871
Non-trainable params: 0
_________________________________________________________________
```

4 Predictions

The results of predicting the test set of the three models is shown in Figure 8. We can see that all the models tend to predict a series that has less variance during the out-of-sample period than the true series. It seems that the ARMA model captures the variance during the COVID period best, and LSTM is worst of doing so. Although the predicted series generated by ARMA model is clearly ”damped”, the fluctuations remains within the confidence interval most of the time.
Table 1 shows the performance measured by mean squared error (MSE) of the three models. The ARMA model has the lowest error, and RNN model has the highest error.
Based on the results, we recommend using the ARMA model.

Figure 8: Results of Predictions



Test Data and ARMA Predictions

Test Data and RNN Predictions

Test Data and LSTM Predictions

Table 1: MSE of Three Models

Model	ARMA	RNN	LSTM
MSE	3.6687×10^{-4}	3.9724×10^{-4}	3.8919×10^{-4}

5 Insights and Findings

We can see that neither of these three models performs ideally in this scenario. These results might due to the fact that our return series are very close to 0. Although these models capture the trend well and remains closely around 0, they do not capture the variance, or risks, in the series.

Another reason might be that the COVID changes the variance of the series so the models tend to underestimate the variance in the out-of-sample period.

The results also indicates that in case of "black swan" event like COVID, simply using forecasting models would not be enough and can lead to great loss in the investment. To prevent the loss the investors need to incorporate risk management methods such as VaR or stop-loss into their strategies.

6 Next Steps

We may want to acquire more observations into our dataset, since deep learning models, especially LSTM, perform better when we have large dataset. We may also continue to try other orders of the ARMA model, and see if changing orders can improve the performance and capture the variance better.

In addition, we might try to use deep learning models directly on price to see if the deep learning model would perform better on predicting the price than predicting the return.