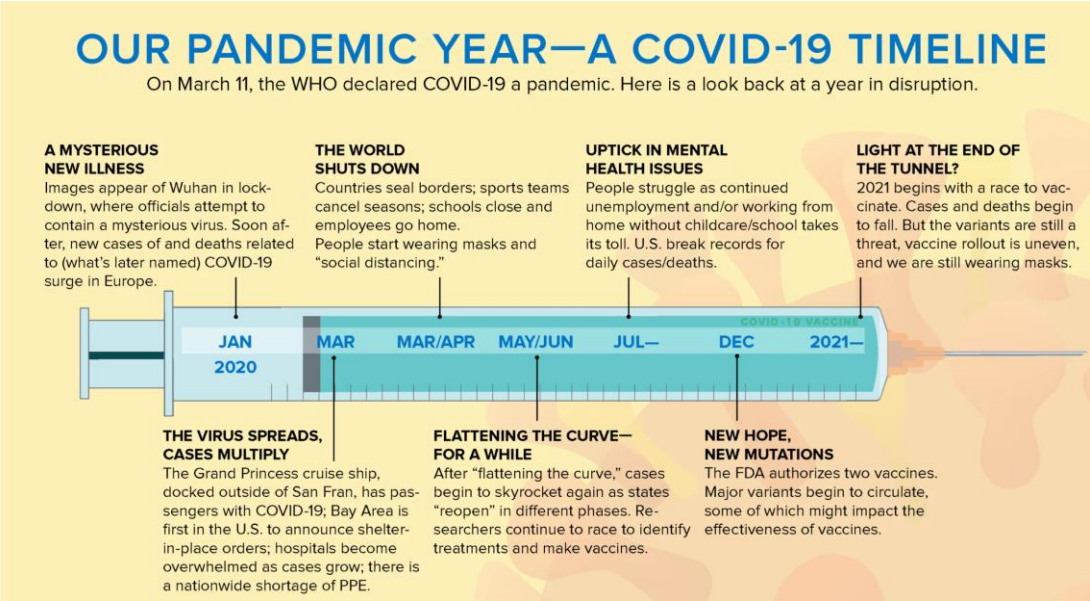
ISE 537 Final Report

Apple and Google Stock Prices Prediction using ARIMA and LSTM model

Chung Ming Wu

Abstract

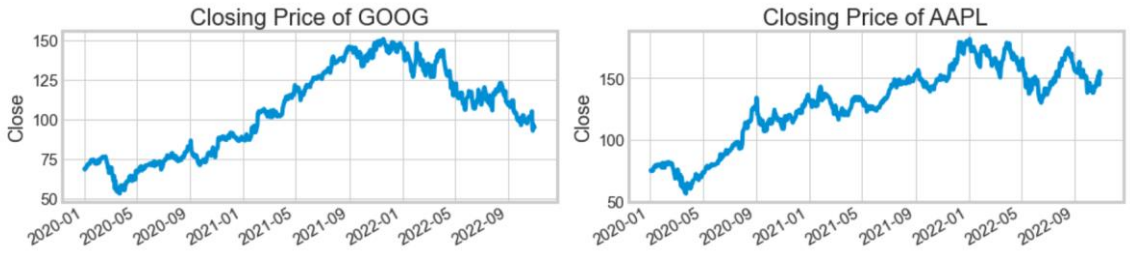
In this report, I will mainly focus on predicting two tech companies Apple (AAPL) and Google (GOOG) stock prices with two different models during the Covid-19 pandemic. One is Autoregressive Integrated Moving Average (ARIMA), and the other is Long Short-Term Memory (LSTM) to analyze the stock trend. The period will mainly focus on is from January 01, 2020, to November 01, 2022, which is starting from the beginning of the Covid-19 outbreak until now.



Covid-19 Timeline https://www.yalemedicine.org/news/covid-timeline

Data

Apply Python's yfinance library to get the stock information, and finally, look at a few ways of predicting the stock price. Like Time-Series (ARIMA) Model and a Long Short Term Memory (LSTM) Model.



Google (GOOG) stock close price (Left); Apple (AAPL) stock close price(right)

The Data Format I use is using Python's library Pandas Data Frame to store the historical closed stock price per stock (as the below picture is the AAPL data).

Date	Open	High	Low	Close	Adj Close	Volume
2020-01-02	74.059998	75.150002	73.797501	75.087502	73.561539	135480400
2020-01-03	74.287498	75.144997	74.125000	74.357498	72.846367	146322800
2020-01-06	73.447502	74.989998	73.187500	74.949997	73.426826	118387200
2020-01-07	74.959999	75.224998	74.370003	74.597504	73.081490	108872000
2020-01-08	74.290001	76.110001	74.290001	75.797501	74.257103	132079200
...	...	...	...	...	...	...
2022-10-25	150.089996	152.490005	149.360001	152.339996	152.087708	74732300
2022-10-26	150.960007	151.990005	148.039993	149.350006	149.102661	88194300
2022-10-27	148.070007	149.050003	144.130005	144.800003	144.560196	109180200
2022-10-28	148.199997	157.500000	147.820007	155.740005	155.482086	164762400
2022-10-31	153.160004	154.240005	151.919998	153.339996	153.086044	97943200

Two models will use in the report, the first is the LSTM model, second is the ARIMA model.

Model 1 - Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) is one type of recurrent neural network which is used to learn order dependence in sequence prediction problems. Due to its capability of storing past information, LSTM is very useful in predicting stock prices. It is because the prediction of a future stock price is dependent on the previous prices.

In this LSTM model, I set **a 30-day window** as timestamps as our feature data(X_train) to build the neural network model. Then, use the open-source machine learning library, **Tensorflow, and Keras**, to set up our LSTM network architecture.

LSTM Layer Selection (Extra Point)

First, define a Sequential model which consists of a linear stack of layers. Then add an LSTM layer by giving it 100 network units. Set the return_sequence to true so that the output of the layer will be another sequence of the same length. Next, add another LSTM layer with also 100 network units. But we set the return_sequence to false for this time to only return the last output in the output sequence. Last, add a densely connected neural network layer with 25 network units. At last, add a densely connected layer that specifies the output of 1 network unit. A summary of our LSTM network architecture is below.

```
Model: "sequential_4"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm_5 (LSTM)               (None, 30, 100)           40800

 lstm_6 (LSTM)               (None, 100)               80400

 dense_5 (Dense)             (None, 25)                2525

 dense_6 (Dense)             (None, 1)                 26

=================================================================
Total params: 123,751
Trainable params: 123,751
Non-trainable params: 0
```

LSTM Layer Parameter

LSTM Model Selection

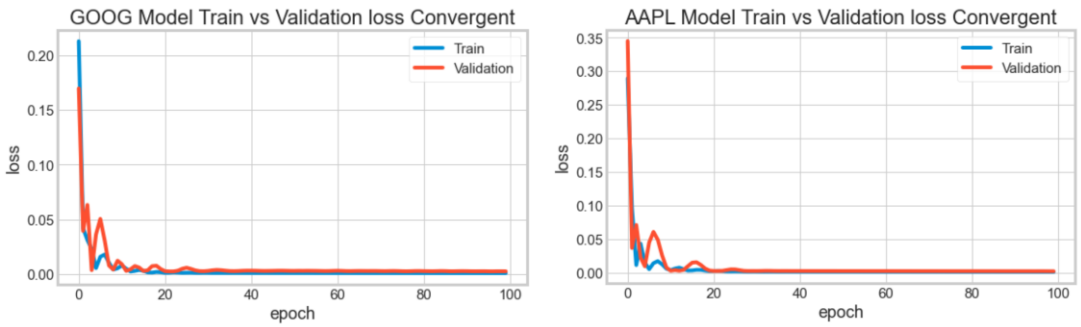
For the optimizer and a loss function, adopt the **"Adam"** optimizer and set the **"mean square error**" as a loss function to fitting it with the training set with batch_size of 256 and run the training for 100 epochs. The reason why using Adam is that it's effective and based on empirical results demonstrate that Adam works well in practice and compares favorably to other stochastic optimization methods. For the loss function using mean square error to minimize the error between the actual stock price and our prediction. Mean squared error is calculated as the average of the squared differences between the predicted and actual values. The result is always positive regardless of the sign of the predicted and actual values and a perfect value is 0.0. The squaring means that larger mistakes result in more errors than smaller mistakes, meaning that the model is punished for making larger mistakes.

LSTM Data Processing

Then separate the dataset by using the train-test split, I use 90% as the train set and 10 % as the test set. Next, use MinMaxScaler to do data scaling between 0 to 1 before we fit the model.

LSTM Convergent

To monitor convergence, using training loss and validation loss together. Set the validation split is 0.33 for both AAPL and GOOG. This can be diagnosed from a plot where the train and validation loss decrease and stabilize around the same point. The results are as the following.

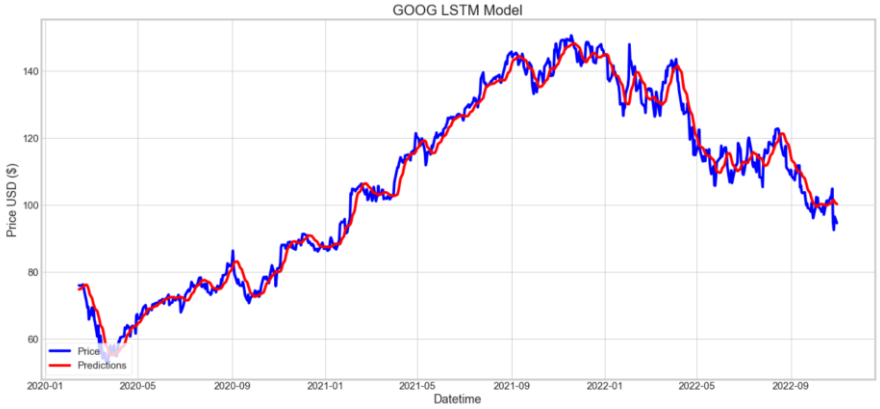


Google (GOOG) Train loss vs Validation loss (Left); Apple (AAPL) Train loss vs Validation loss (right)

LSTM Evaluation

To evaluate the result, I apply the Root Mean Square Error (RMSE) metric to examine the model performance. For AAPL's train set RMSE is 4.81, and the test set RMSE is 5.39; for GOOG's train set RMSE is 3.48, and the test set RMSE is 4.26. In the problem, we use Actual stock price to calculate RMSE instead of %, which means that both our training and testing results are good. From the resulting chart below, we can see the predicted stock prices follow the trend of the real stock prices closely. This shows the effectiveness of the LSTM to work very well.

Apple (AAPL) stock price vs prediction(above); Google (GOOG) stock price vs prediction (below)



Model 2 - Time-Series Model (ARIMA)

ARIMA model is a class of statistical models for analyzing and forecasting time series data. It explicitly caters to a suite of standard structures in time series data, and as such provides a simple yet powerful method for making skillful time series forecasts. For ARIMA models, a standard notation would be ARIMA with p, d, and q, where integer values substitute for the parameters to indicate the type of ARIMA model used. The parameters can be defined as:

- p: The number of lag observations in the model; also known as the lag order.
- d: The number of times that the raw observations are differenced; also known as the degree of differencing.
- q: The size of the moving average window; also known as the order of the moving average.

ARIMA Data Stationarity

Before fitting into the ARIMA model, we need to check the data stationary. To check whether the data is stationary, we will use the Augmented Dicky Fuller (ADF) test. ADF is a statistical unit root test that tests the null hypothesis that $\alpha=1$. Alpha is the coefficient of the first lag on Y. For each stock, GOOG and AAPL, use 4 different values to do the ADF test. **(i) Close price, (ii) Close price difference, (iii) take the log of the Close price, and (iv) the log of the Close price difference**. From the AAPL result, it is evidence shows that (i) & (iii) t-Statistics are greater than the critical values, for which the p-value > 0.05, we fail to reject the null hypothesis meaning that the time series data is non-stationary. There are various methodologies to convert non-stationary data to stationary data but now in this report, take the log of the "Close" price as (ii) and (iv) did. This time, when applying the ADF test, the t-Statistics is less than the critical values, so p-value < 0.05, we reject the null hypothesis meaning that the time series data is stationary.

ARIMA Model Selection

We can take any combination of values for the ARIMA model and use both AIC and BIC criteria to select the ARIMA (p, d, q) model within the class of $p \leq 8$ and $q \leq 8$. Then, we are selecting values of the ARIMA model using the **auto_arima** function from the pmdarima package. The values are chosen in such a way that the AIC or BIC score is minimized. For AAPL, when using **(ii) Close price difference**, AIC suggests model ARIMA (5,0,2); however, BIC suggests model ARIMA (0,0,0). In this case, AIC is best for prediction as it is asymptotically equivalent to cross-validation. BIC is best for an explanation as it allows consistent estimation of the underlying data-generating process. Thus, we take minimized AIC as the best model to obtain the optimal parameters (5,0,2). If we work with **(iv) the log of the**

Close price difference in AAPL, AIC suggests model ARIMA (2,0,5); however, BIC suggests model ARIMA (1,0,0). We obtain the optimal parameters (2,0,5).

For GOOG, when using **(ii) Close price difference**, AIC suggests model ARIMA (2,0,7); however, BIC suggests model ARIMA (0,0,0). Obtain the optimal parameters (2,0,7). If we work with **(iv) the log of the Close price difference** in GOOG, AIC suggests model ARIMA (7,0,1); however, BIC suggests model ARIMA (1,0,0). This time we obtain the optimal parameters (7,0,1).

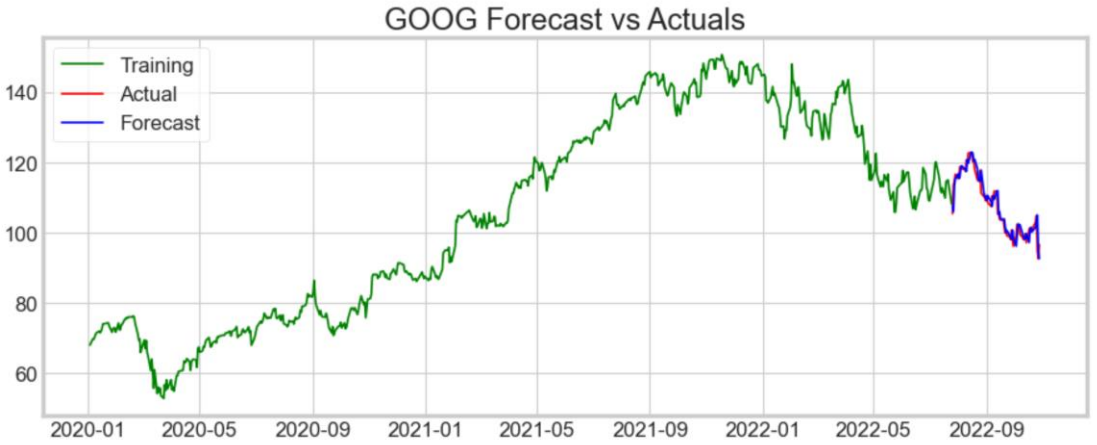
ARIMA Evaluation

According to the model summary in both **(ii) and (iv),** the model meets the condition of independence in the residuals (no correlation) because the p-value of the Ljung-Box test (Prob(Q)) is greater than 0.05, so we cannot reject the null hypothesis of independence, but we cannot say that the residual distribution is homoscedastic (constant variance) because the p-value of the Heteroskedasticity test (Prob(H)) is smaller than 0.05.

When using AAPL's **(ii) Close price difference**'s train set RMSE is 2.63 and the test set RMSE is 3.39; AAPL's **(iv) the log of the Close price difference**'s train set RMSE is 0.02 and the test set RMSE is 0.02. Both are predicting very well as below chart.

Apple (AAPL) stock price vs ARIMA prediction using (ii) Close price difference

For GOOG's **(ii) Close price difference**'s train set RMSE is 2.01 and the test set RMSE is 2.56; GOOG's **(iv) the log of the Close price difference**'s train set RMSE is 0.02 and the test set RMSE is 0.025.



Google (GOOG) stock price vs ARIMA prediction using (ii) Close price difference

Model Comparison LSTM vs ARIMA

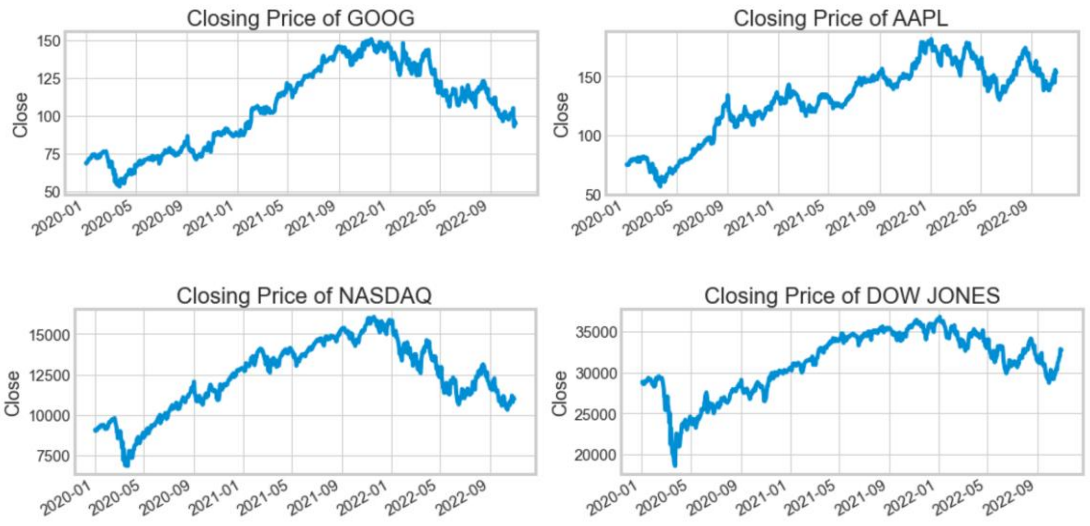
From the model evaluation results above, we presented and compared two different algorithms for time series prediction. As expected, we can see the predicted stock prices follow the trend of the real stock prices closely no matter using LSTM or ARIMA model. When comparing the out-of-sample performance, both models are still tied. There is no clear winner, and each algorithm has its advantages and limitations.

ARIMA is a powerful model as we saw it achieved the best result for the stock data. Besides, only using stock diff can pass the ADF test. One challenge is that it might need careful hyperparameter tuning and a good understanding of the data.

LSTM-based recurrent neural networks are probably the most powerful approach to learning from sequential data and time series are only a special case. The potential of LSTM-based models is fully revealed when learning from massive datasets where we can detect complex patterns. Unlike ARIMA, do not rely on specific assumptions about the data such as time series stationarity or the existence of a Date field. Moreover, data needs scaling before fitting the models compared with the ARIMA model. One disadvantage is that LSTM-based RNNs are difficult to interpret, and it is challenging to gain intuition into their behavior. Also, careful hyperparameter tuning is required to achieve good results.

Financial Interpretation and other factors

By plotting the time series data, we observe both AAPL and GOOG stock trends have very similar behavior between 2020-01 to 2022-01 as the stock market index like NASDAQ and DOW JONES. Covid-19 appeared in Wuhan, China in Jan 2020. At that time, we still had limited knowledge about Covid-19. Thus, stock markets across the world suddenly crashed after growing instability due to the COVID-19 pandemic during Feb- Apr 2020. Fortunately, people started to take Covid-19 vaccines like Pfizer-BioNTech, Moderna, and Johnson & Johnson. Therefore, the stock market surged between Apr 2020 and Jan 2022. For both AAPL and GOOG even hit a new record high during this period. Until now, AAPL's Market Cap is around $2.24 trillion; GOOG's Market Cap is around $1.23 trillion.



Unfortunately, the global market faced high inflation issues from the beginning of 2022 until now. The most recent CPI report showed the index increased by 0.4% from September 2022 to October 2022. Over the previous 12 months ending in October, the index rose 7.7%, indicating

average prices for common goods and services are rising more than usual. Typically, Americans can expect the CPI to increase between about 1% and 4% each year, based on data from the last couple of decades. During the pandemic, the annual inflation rate stayed below 4% until April 2021. That month, the CPI increased 4.2% over the previous year and continued to rise until it appeared to peak in June 2022, when the year-over-year increase was reported at 9.1%

In this situation, people started to slow down buying tech devices. Further, it will directly impact AAPL and GOOG's revenues. Thus, Apple is reportedly slowing hiring and spending in 2023 amid rising inflation and interest rates, according to Bloomberg. Apple outmaneuvered the economic turbulence caused by the global chip shortage, COVID shutdowns, and the war in Ukraine for months. But the trio of disasters finally caught up to the company and is expected to cut $4 billion to $8 billion out of Apple's Q3 revenue. Google, the tech giant also slow hiring for the rest of the year, according to an email sent to staff from CEO Sundar Pichai. The revenue growth in the September quarter was 6% compared to 41% a year earlier. Other than one period early in the pandemic, it's the weakest period for growth since 2013. Therefore, both GOOG and AAPL were weak as other tech companies in 2020.

Conclusions

To summarize, both ARIMA and LSTM can be great tools for stock price prediction during the Covid-19 period. Different models might be more suitable for each prediction at different time scales. However, this is important to note that the predicted stock prices shall not be used as a solely definitive guide to making an investment decision without further analysis. This is because the prediction is only based on the historical price movement which usually won't be the only factor that affects future price movement. The main limitation of using any machine learning algorithm in predicting stock prices is that we can only test the historical data, but the price movement does not necessarily follow the historical trend in various unforeseen circumstances. That's the reason a further fundamental/market analysis is required here to support our investment decision-making. The findings can provide a good reference for the future selection of prediction models.

Reference

Stock Prices Prediction Using Long Short-Term Memory (LSTM) Model in Python.

Stock price prediction using ARIMA Model.

Stock market forecasting using Time Series analysis With ARIMA model

How to Interpret ARIMA Results

How to confirm convergence of LSTM network?

Forecasting Time Series with Auto-Arima

Advanced Time Series Modeling (ARIMA) Models in Python

Time Series Part 3 - Stock Price prediction using ARIMA model with Python

How to Diagnose Overfitting and Underfitting of LSTM Models