Stock Price Prediction via Machine Learning
James Yan (50118541)

Problem Statement

With the rising trend of big data and digital technology, stock market prediction has become an active area of research. Since companies have to file updated financial reports to the SEC, stock price data is publicly available online. On paper, having publicly available data could be helpful to predict the risks and opportunities associated with the company and its business. However, the stock market's chaotic system makes its uncertainty rather large for prediction. Due to a variety of factors, we are currently not sure how true the Efficient Marketing Hypothesis is for stocks, considering that they may not reflect past historical data due to coincidental anomalies (such as financial disasters to give an example).

Though the risk of the market is quite high, the makes it a ripe epicenter for making money. In retrospect, if a potential retail trader, investor, or even a growing company could predict the price of a stock accurately, one could reap significant profit yields.

Trying to predict the stock market is impossible, but certain strategies can help minimize the uncertainty of stock market prediction, and machine learning could be one strategy.

This project attempts to apply machine learning algorithms, particularly in regression and/or classification, to predict a sample set of stocks using past financial data on its stock price, specifically big and more stable companies. This project will hopefully explore a base strategy for predicting future prices for stocks.

Datasets

This project will plan to use data from Yahoo! Finance, a media property that provides up-to-date information for company stock quotes. It is more than enough for this project since it only provides daily stock data price.

The most recent 10 years will be used since future stock prediction will be most accurate. It would not make sense to use data before the recent financial disaster of 2008 since the stock price environment would be much different than the most recent decade.

We will be looking at big companies, particularly the S&P 500, as a base dataset since they are much more stable in price fluctuation, so it would be more effective to set our base strategy for big companies.

Methodology

The data will first go through pre-processing, starting with extracting the features (Adj. Closing Price and Date). Then the data will be standardized to control the variability of the price data.

Since historical (time-series) price data mostly involves numerical data, appropriate algorithms would mostly be regression based. However, linear regression would not work since most stock data is not linear-trending. Instead, we will use SVR (Support Linear Regression) as demonstrated by Henrique, Bruno Miranda, et al. [1].

SVR combines linear regression and support vector machines by attempting to find a function $f(x)$ that is at most ε-deviation from the target data. The incorporation of ε will attempt to minimize the noise

Minimize $L(w) = \frac{1}{2}\|w\|^2$ s.t.

$$|y_i - f(x_i)| \leq \varepsilon$$

Model for SVR.

We will use the radial basis kernel function $K(x, x')$, as defined below:

$$K(x, x') = \exp(-\frac{\|x - x'\|^2}{2\sigma^2})$$

We are using this particular kernel because it can account for the variation and noise in stock price the best. Sigmoid and linear kernels will not fit the data very well due to the stock volatility, and polynomial kernels are heavily sensitive to its degree parameter.
Also, I will be using a LSTM (long short-term memory) neural network, a subclass of RNNs, as mentioned in Roondiwala, Murtaza & Patel, et al. [2]. The architecture will consist of two hidden layers with 40 nodes, run under 5 epochs. Since they can remember trained information for long periods of time, it will be ideal for stocks.
This model will be implemented using the Python Keras package as follows:

```
model = Sequential()
model.add(LSTM(units=40, return_sequences=True,
input_shape=(x_train.shape[1], 1)))
model.add(LSTM(units=40))
model.add(Dense(1)) # outputs one value
```

We will split the data 80-20: 80% will be used for training the model, while the 20% most recent data will be used for validation/testing.

Results

To measure the accuracy of the models, a general loss function $\mathrm{L(w)}$ for regression, as measured by RMSE, will be used to observe how accurate the predicted price is compared to the actual price:

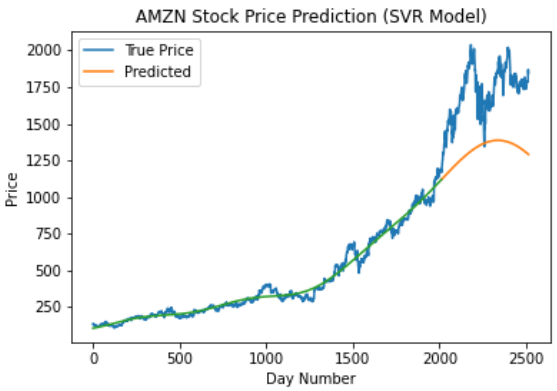
$$RMSE = \sqrt{\frac{1}{N}\sum(\hat{y} - y)^2}$$

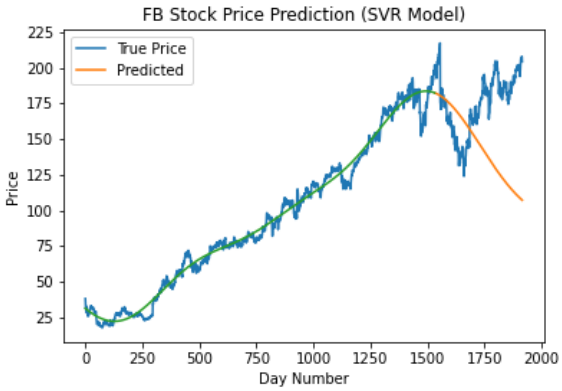
A sample set of 5 stocks from the S&P 500 were tested via SVR, primarily FB, AMZN, AAPL, GOOG, and WMT.

Ticker	*RMSE (SVR)*	*RMSE (LSTM)*
FB	48.1844613030269	8.417286796630298
AMZN	425.1585946989589	169.42922794035422
AAPL	54.2598303282391	10.542643592556692
GOOG	95.33713598003776	51.69869084391755
WMT	10.654235077441264	2.717556974502158

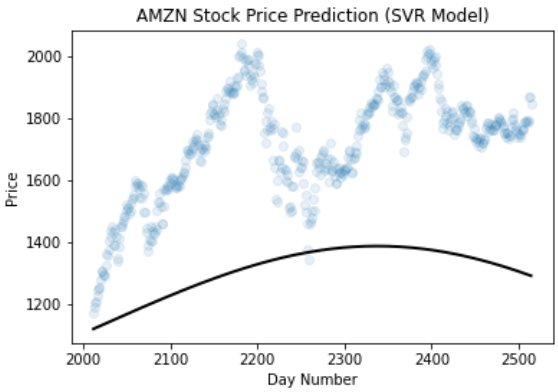
Based on the RMSE, it can vary depending on how much noise and/or deviation there is from the test d ata.

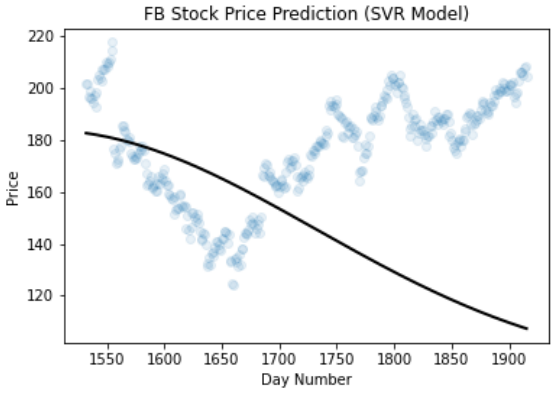
From the visualizations, the results of the fitted line appear to generalize the stock price (specifically, the training data) trend really well, as demonstrated in the two example graphs.



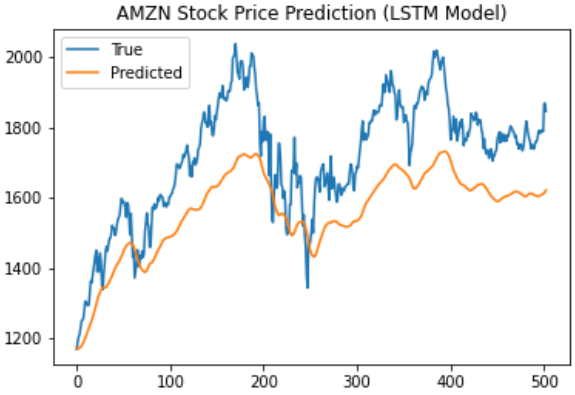


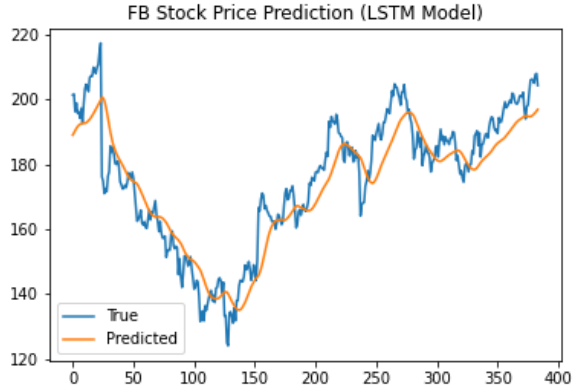
However, there appears to be a large margin of error when zooming into the test data set.





As for LSTM, its RMSE is quite low, as it accounts for the variation and noise much more than SVR, as demonstrated in the two example graphs (only the predicted test data set is shown here). However, sometimes it is not centered.





Combined Model

One way to make a new model out of the two pre-determined models is to combine them to make a new model based on both the model's predicted values. We would need to do so since we would like to obtain a method that could generalize the noise and variance for stock prediction.

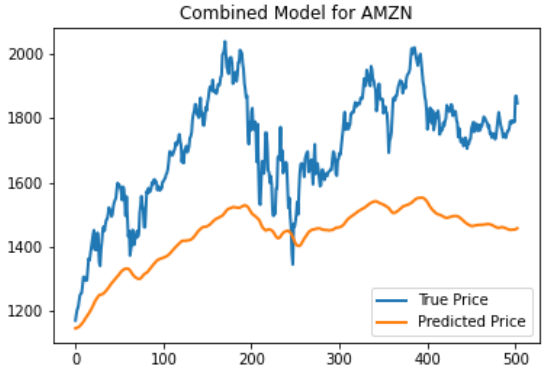
For now, a simple combined model would be to take the average of the predicted values for both models:

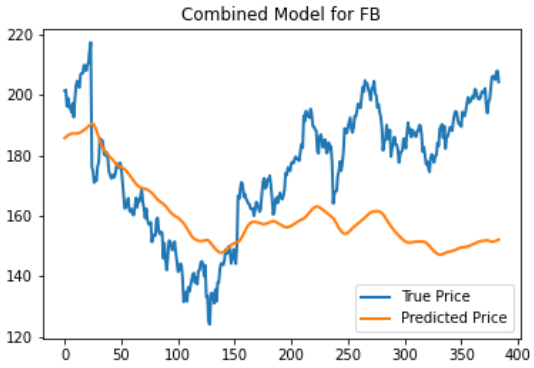
$$New_{Model} = \frac{SVR_{pred} + LSTM_{pred}}{2}$$

From this, we have the following loss results:

Ticker	*RMSE (Combined Model)*
FB	8.417286796630298
AMZN	169.42922794035422
AAPL	10.542643592556692
GOOG	51.69869084391755
WMT	6.154708259289999

Based on the results, the RMSE is pretty much a middle ground between the high values in SVR and the low values in LSTM. The two sample graphs also have a medium amount of noise as well as a medium amount of accuracy compared with the two separate models.





Based on both results, this would be the best model for predicting future stock prices.

Future Steps

Future steps for the project, considering the results, would be to either make a more versatile ensemble model to obtain a method that could generalize the noise and variance for stock prediction. This can be done by either adding another well-generalized model, or (a) improving previous model(s).

References

[1] Henrique, Bruno Miranda, et al. “Stock Price Prediction Using Support Vector Regression on Daily and up to the Minute Prices.” The Journal of Finance and Data Science, vol. 4, no. 3, 2018, pp. 183–201., doi:10.1016/j.jfds.2018.04.003.

[2] Roondiwala, Murtaza & Patel, et al. (2017). Predicting Stock Prices Using LSTM. International Journal of Science and Research (IJSR). 6. 10.21275/ART20172755.

[3] Google Code Folder: https://drive.google.com/open?id=1MOfD_RhAIC7VeO4Fri01OFr35DDutmYM