Ryan Lin

Stock Prediction Using LSTMs With Sentiment Analysis

Exploring Machine Learning's Capability to Predict Complex, Volatile Data

Introduction

As the field of machine learning increases in popularity, many day-to-day processes find themselves dependent on some form of a predictive model. Whether it be targetted advertising through a deep belief network-logistic regression model (DBNLR) or optimized 1-day shipping using recurrent neural networks, most of the conveniences of the modern world are possible because of machine learning. As this new field of study works its way into our daily lives, the question arises: how powerful is machine learning, and to what degree can we rely upon it? To explore this question, we will take a look at perhaps one of the most popular use cases for machine learning, the prediction of stocks.

The challenge of predicting stocks through machine learning lies in the sheer number of factors that can affect the valuation of a company. The variables attached to stocks are their opening price, closing price, daily high, volume, and adjusted closing price on given dates. Predicting stocks using machine learning is not a novel task, and has been performed in a variety of ways. However, very few of these methods are able to account for the various background effects that can change stock value. For example, even a simple change in weather could result in a shortage of a certain resource, and thus an increase in the valuation of the resource. This could in turn lead to surges in prices that cannot be accounted for by a human (at least, not beforehand). Stocks are also dependent on volatile factors such as public opinion, which is a seemingly non-periodic variable that cannot be accounted for. In addition, many of these variables are not dependent on past combinations or patterns, further suggesting that there are many unobservable effects at work. The question at hand is: just how well can machine learning do its job? Can it account for all these various factors with just a few predictor variables where a human cannot? If so, is it reliable?

To begin with, I first used a single-variable Long Short-Term Memory model (LSTM) on the closing cost variable itself, fitting the model to the time series and training it accordingly. I then experimented with training based on multiple variables provided by the API I used (namely “Volume”). The goal of this was to see if the addition of other easily-accessible variables in the data would increase model performance, as well as to assess how LSTMs handle multiple predictors and compare to the corresponding single-variable model (both trained and tested on the same data). Finally, I introduced a new variable to capture the public’s opinion of a given company (through sentiment analysis of related media) and compared the results of each approach.

Methodology and Model Selection

There are several methods to approximate the progression of time series data (like stocks). In the finance field, the two most popular models for this task are the **Autoregressive Integrated Moving Average (ARIMA)** model and **Long Short-Term Memory (LSTM)** model.

Costs vs. Benefits of ARIMA

ARIMA models are inexpensive to gather data for. All that is required is prior data of the time series. ARIMA also performs extremely well on short-term forecasts and can model non-stationary data, which is the term for data with means and variances that change over time (Bora).

However, they are computationally expensive and perform poorly when predicting turning points. In addition, the performance of ARIMA decreases drastically for long-term forecasts, partially because ARIMA is unusable for seasonal time series like stocks in the long term, which is a large drawback for the purposes of this project (Bora). Furthermore, ARIMA models are only trained on one variable, which limits the flexibility of factoring in more predictors.

Costs vs. Benefits of LSTM

LSTMs are able to self-correct the back-propagating error within memory cells. As such, LSTMs can handle the potential noise that can be encountered in long time lag problems, like in our case with stocks (Hochreiter). More importantly, LSTMs can be trained upon multiple predictors, which provides for more flexibility down the line that ARIMA cannot.

Unfortunately, there are still a few drawbacks to using an LSTM. These types of models are subject to overfitting and are also computationally expensive to train (Siami-Namini).

Final Model Selection

After analyzing the drawbacks of each type of model, the final decision is to use the LSTM due to its better performance in long-term forecasts and factor in multiple predictors. Another reason an LSTM was chosen was that “The average reduction in error rates obtained by LSTM is between 84 - 87 percent when compared to ARIMA” (Siami-Namini).

In an attempt to answer the question of whether machine learning is capable of reliably predicting stocks, several approaches will be compared through their RMSE and accuracy when predicting whether a stock increases or decreases in value on a given day. A single-variable LSTM, a multi-variable LSTM, and a multi-variable LSTM that is supplemented by sentiment analysis (this final approach would supposedly account for unpredictable external factors such as natural disasters or public opinion).

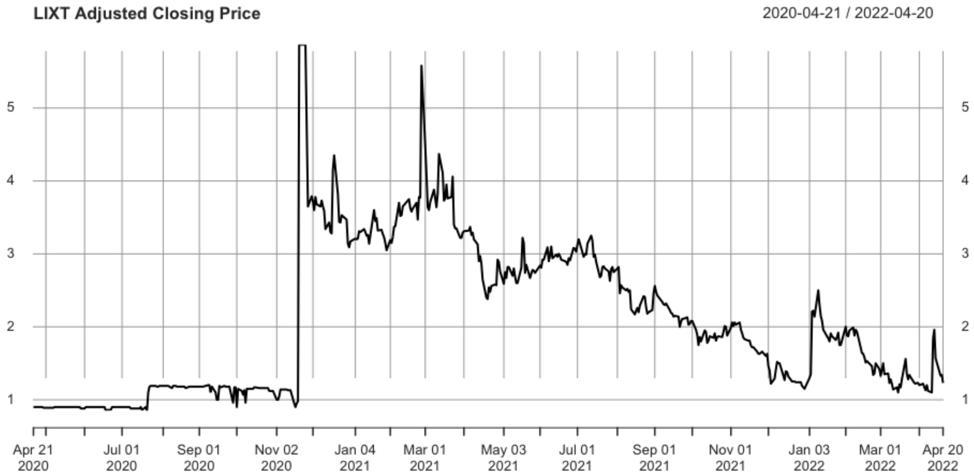
Sentiment Analysis

To account for the possible variability of the public opinion of a company, sentiment analysis was incorporated into the model. Using the R library “rvest,” The program scrapes Google News and queries the news articles that relate to the given company on the given dates in the dataset. It then conducts sentiment analysis on the top results, averages them, and stores the value. The sentiment values are obtained using the “SentimentAnalysis” library in R (training and utilizing a sentiment analysis model built from scratch would be far out of the scope of this research). This information is then used as a predictor for the final model in the exact same way one adds another predictor to a multi-variable LSTM.

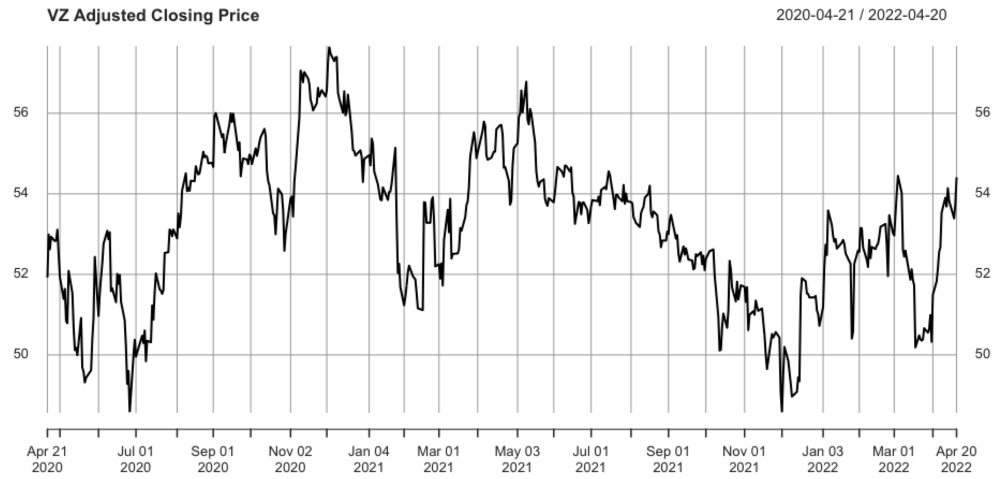
Data Description

Several stock datasets were pulled from Yahoo finance through the R package “quantmod.” The program first acquires the current date and queries the stock data of a specific company from the previous five years.

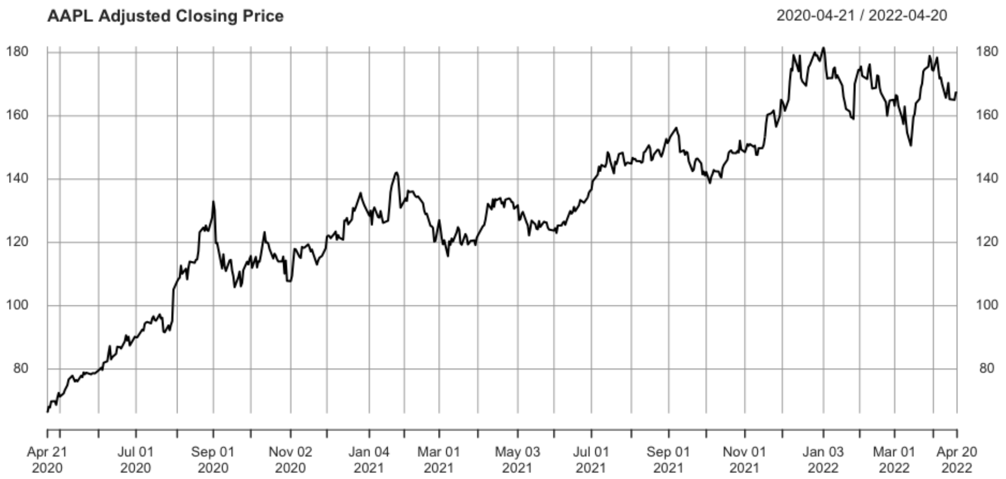
These specific companies were selected based on several defining characteristics of their motion throughout the years.



Lixte Biotechnology Holdings (LIXT) was chosen because of its characteristic volatility (take for example the spike in late-2020).



Verizon (VZ) was chosen for its stability (when compared with other companies).



Apple was chosen because of its steady (albeit astounding) rate of growth.

The goal when selecting these companies was to cover as diverse a range of stocks as possible. A stock that is unpredictable, a stock that is stable, and a stock with a clear general trend. Apple in particular also has the added complexity of seasonal trends, seeing as its value spikes up with the periodical release of new iPhones every year.

Data Analysis

Data Preparation

To train the model, there is much pre-processing of the data to be done.

Firstly, the data must be transformed such that the sequence is the difference between each term of the previous sequence. For example, a series of (1,3,5,7,9) would be mapped to a *difference series* of (NA,2,2,2,2).

Then, the difference series is lagged by a certain time period (1 day was used in the program). The LSTM will use the lagged time series (past) to try and predict the actual time series (future).

Next, all the data must be standardized and then scaled so that the weights assigned to the model are of the proper magnitude. Data are standardized by subtracting the minimum from the entire dataset (so the new minimum is 0) and dividing by the range. The data is then scaled by mapping it to a (-1,1) range.

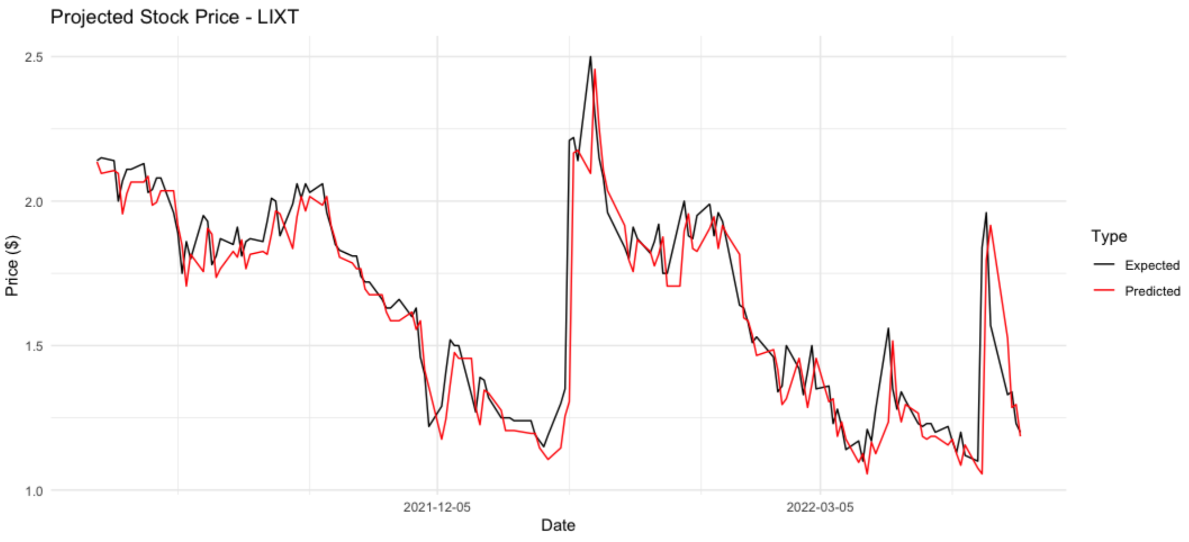
Then, the scaled data must be reshaped into an Nx1xF matrix where N is the number of entries in the time series and F is the number of features used to train the model. The figure to the right is a visualization of the reshaped training data (the features being used are “Adjusted” and “Volume”).

Running and Scoring the Model

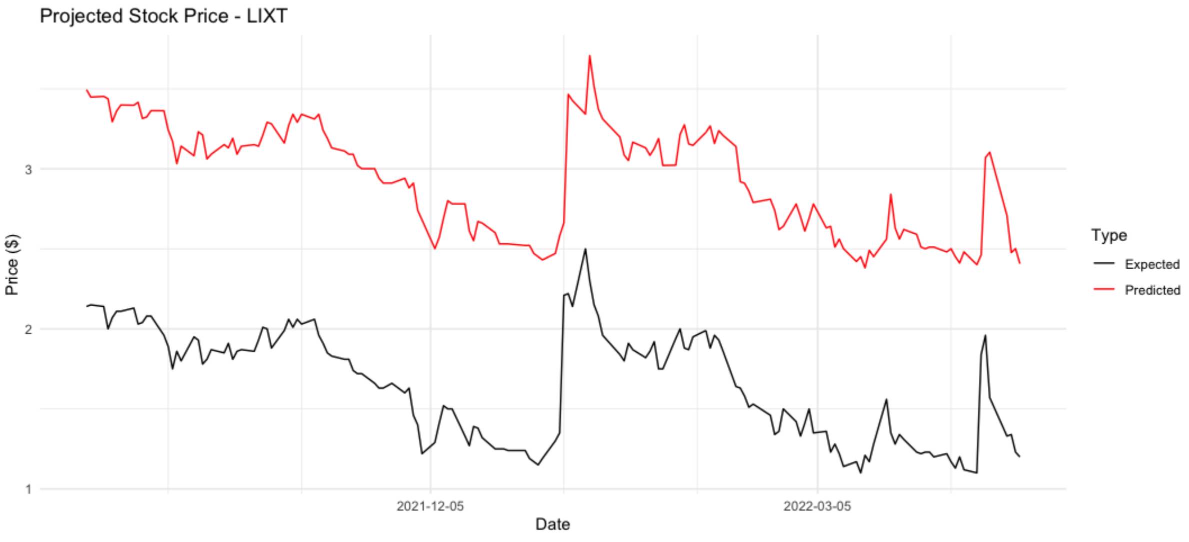
After being split into a training and test set (a 70/30 split was used), the training data is fed into the LSTM model. Upon the conclusion of the training phase, the model is fed the test set one entry at a time and predicts the corresponding output (of course, we would have to invert the scaling that we initially performed). Each model is then scored by its Root Mean Squared Error (RMSE) and its accuracy when predicting whether the value of the stock will increase or decrease on each of the given dates in the test set.

Results

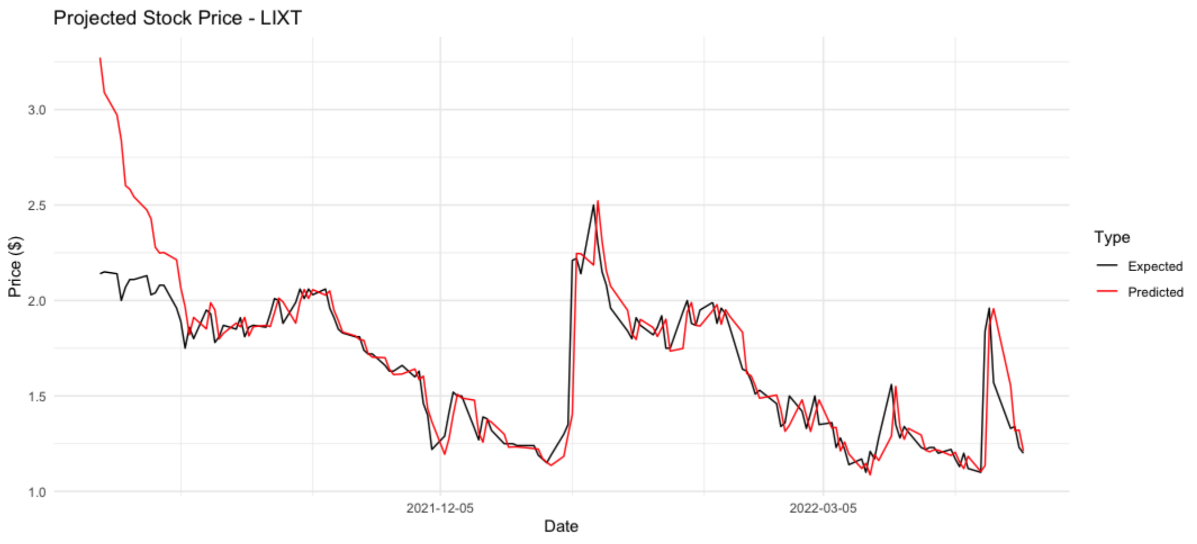
Lixte Biotechnology Holdings Inc



Prediction w/ Single-Variable LSTM

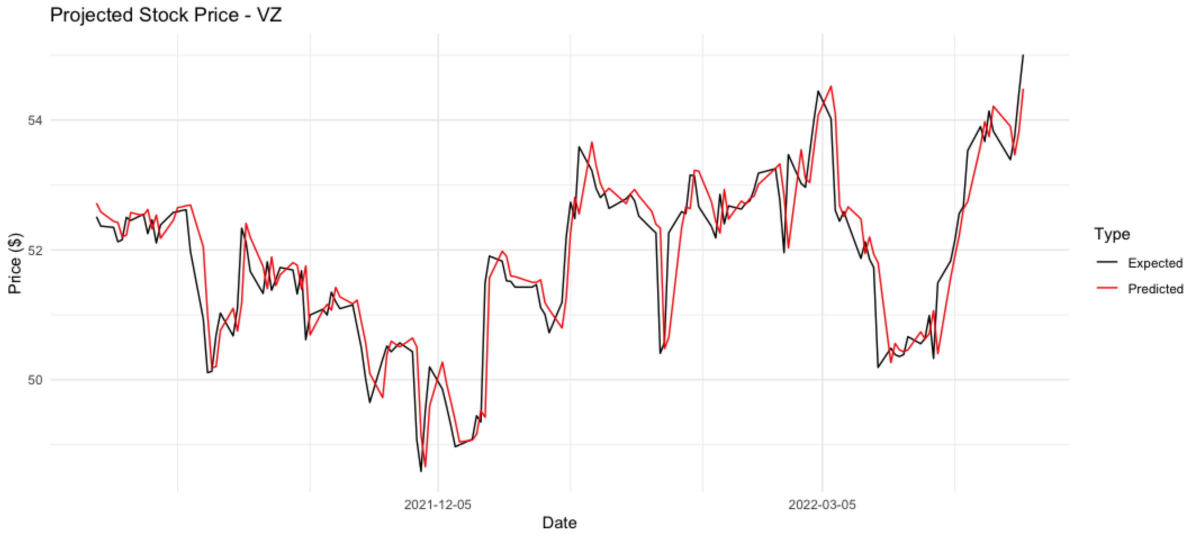


Prediction w/ Multi-Variable LSTM

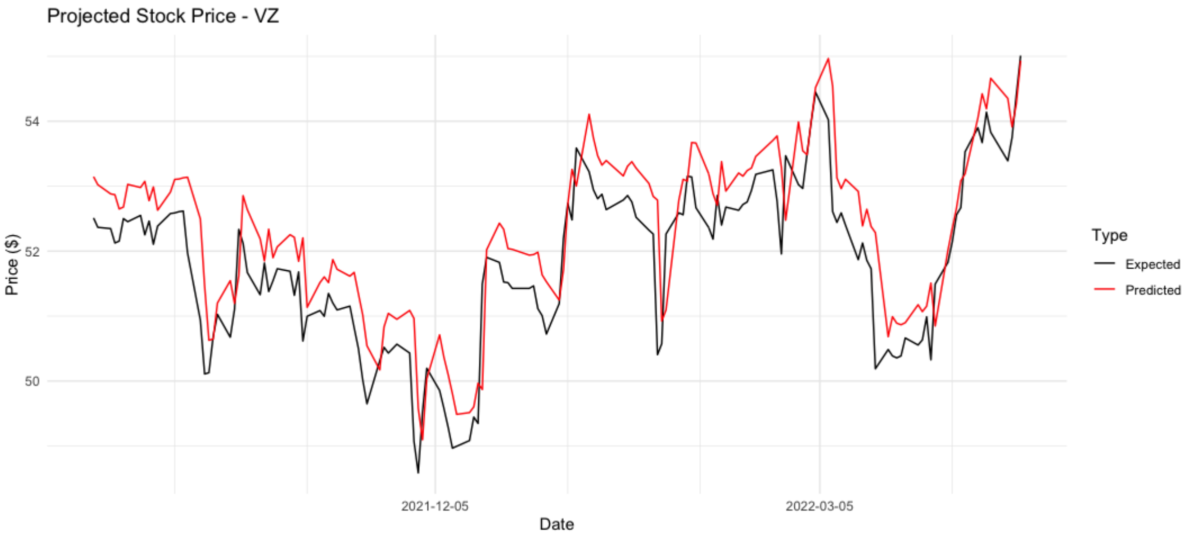


Prediction w/ Sentiment Analysis + LSTM

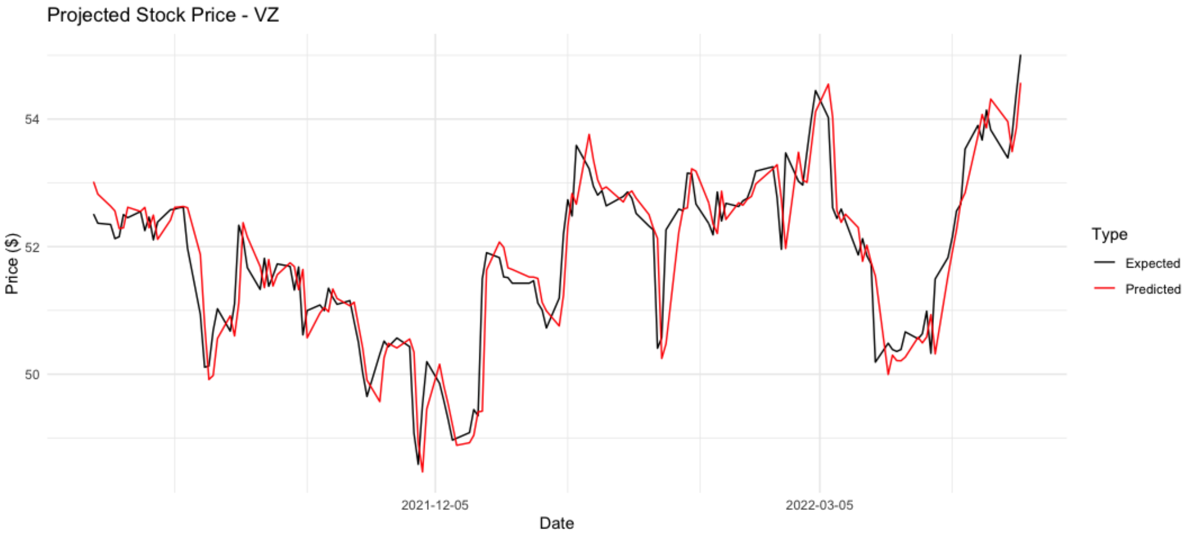
Verizon Communications Inc



Prediction w/ Single-Variable LSTM

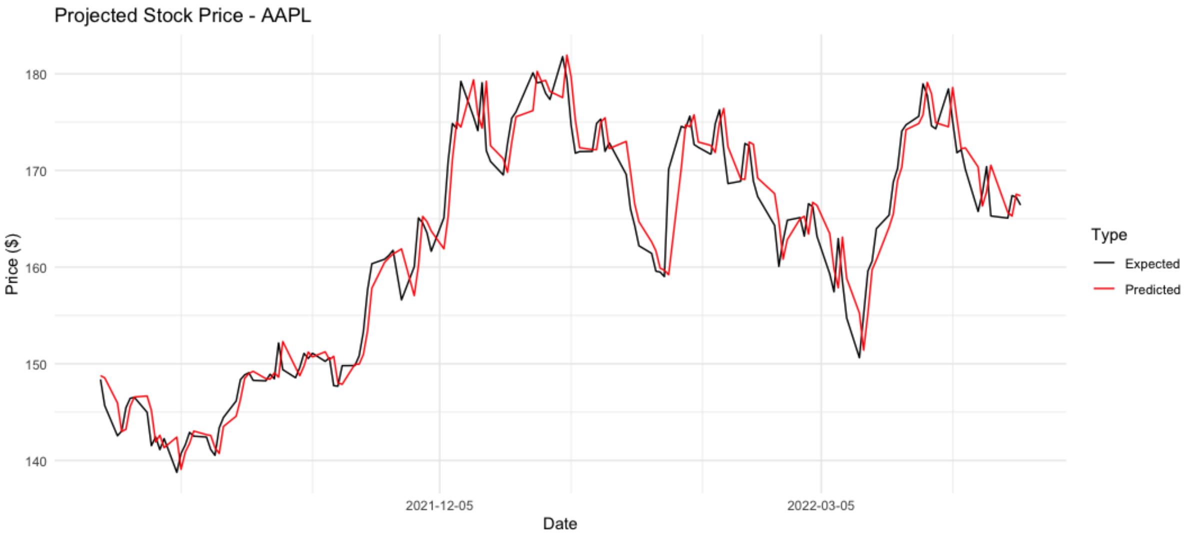


Prediction w/ Multi-Variable LSTM

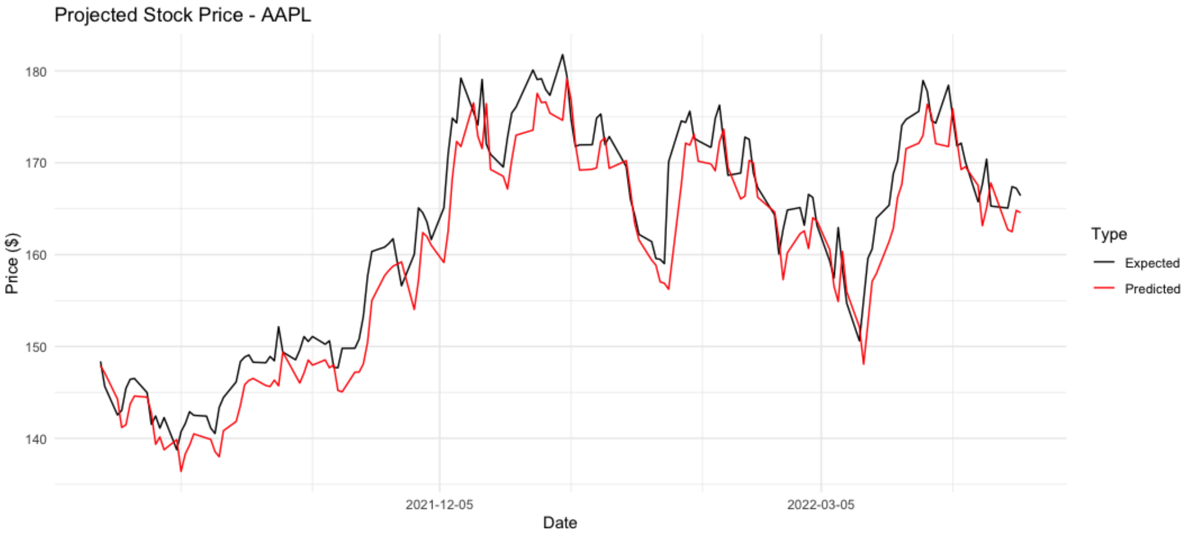


Prediction w/ Sentiment Analysis + LSTM

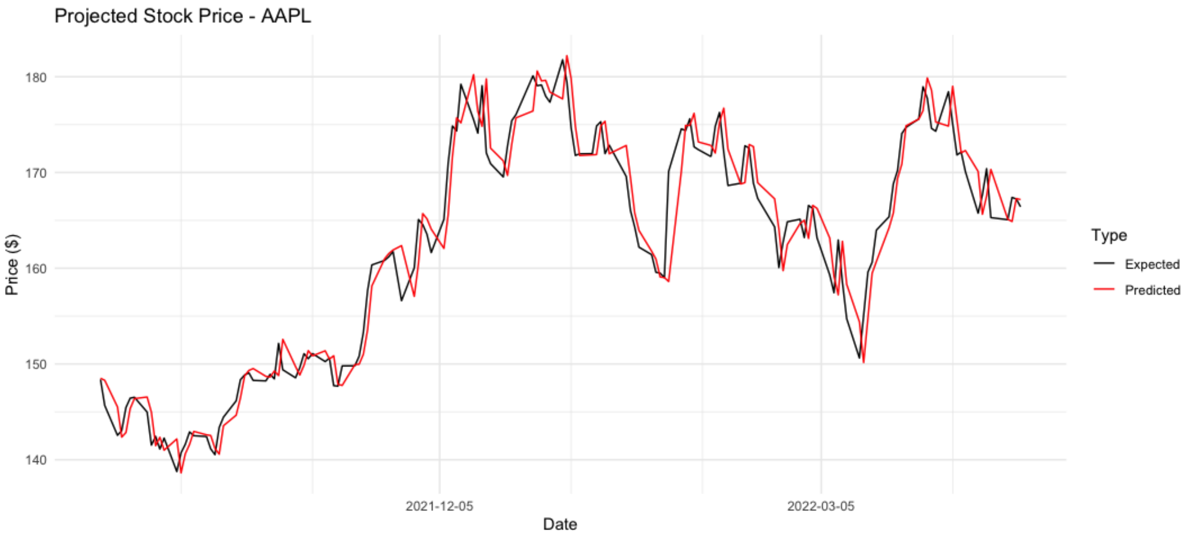
Apple Inc



Prediction w/ Single-Variable LSTM



Prediction w/ Multi-Variable LSTM



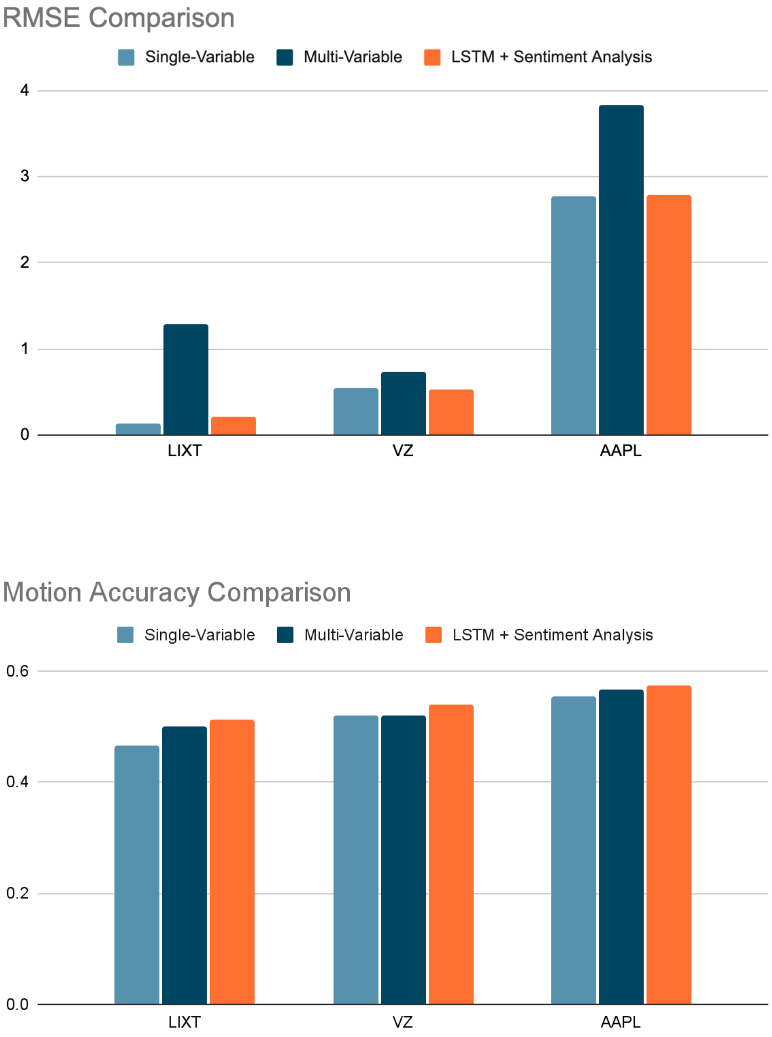
Prediction w/ Sentiment Analysis + LSTM

The Multi-Variable LSTM model seems to visibly perform better than the Single-Variable for LIXT and AAPL, but not for VZ. This can be seen from the Single-Variable model's tendency to deviate from the expected value at several turning points along with the graphs. To quantify these findings, the models' scores are recorded in the tables below.

Single-Variable LSTM			
Company Name	**Symbol**	**RMSE**	**Motion Accuracy**
Lixte Biotechnology Holdings	LIXT	0.1381291	0.4667
Verizon Communications Inc	VZ	0.5368389	0.52
Apple Inc	AAPL	2.770266	0.5533

Multi-Variable LSTM (adding "Volume")			
Company Name	**Symbol**	**RMSE**	**Motion Accuracy**
Lixte Biotechnology Holdings	LIXT	1.286561	0.5
Verizon Communications Inc	VZ	0.7350073	0.52
Apple Inc	AAPL	3.831567	0.5667

LSTM w/ Sentiment Analysis			
Company Name	**Symbol**	**RMSE**	**Motion Accuracy**
Lixte Biotechnology Holdings	LIXT	0.2169114	0.5133
Verizon Communications Inc	VZ	0.5257501	0.54
Apple Inc	AAPL	2.782951	0.5733



Visual comparisons of RMSE and Accuracy of Models

As the charts show the RMSE for the model that incorporates sentiment analysis remains similar to that of the single-variable LSTM. When comparing the accuracy, we can see that the sentiment analysis has a rating that is greater than or equal to that of the two other models for all the tested cases.

Conclusion

Before the investigation, the assumption was that the incorporation of sentiment analysis in a predictive model would increase its performance. Results show that using sentiment analysis and an LSTM in conjunction does indeed marginally improve the success when predicting motion accuracy while retaining a similar RMSE value. Using sentiment values of article titles that relate to a company on a given date seems to overcome the tendency that the normal multi-variable LSTM has to be offset from the true value (this trend is exhibited in the multi-variable LSTM predictions of all three companies).

A common rule of thumb in finance is that 60% accuracy will achieve meaningful returns when investing. Unfortunately, while the models did approach higher accuracy (while maintaining RMSE) as the experimentation continued, the performance still never exceeded that threshold. Further steps to continue this investigation could involve a more in-depth approach to the sentiment analysis. Due to time constraints and limitations on processing power, the model was only able to conduct sentiment analysis on the top resulting article when scraping Google News (if we wanted it to complete processing in a reasonable time). This means that our “sentiment” variable would be extremely biased. In addition, a point of concern is the performance of the sentiment analysis library itself. In various cases when testing, it was apparent that for some articles with clear positive or negative connotations in the headlines, the sentiment analysis library would not know how to accurately give an assessment (resulting in a default sentiment value of 0.00/neutral). With more processing power and a better-performing sentiment analysis model/library, it is reasonable to assume that the model could have reached the 60% accuracy threshold.

The question this experimentation hoped to answer was whether machine learning could accurately predict a value that is dependent on a seemingly endless number of variables. Overall, the results do indeed show that, when given the right information, a machine learning model’s performance can improve marginally. With just two variables (sentiment and previous adjusted closing prices), an LSTM was able to account for a substantial portion of the variability in the data.

Bibliography

Allaire, J.J. *R Interface to Keras • Keras*, https://keras.rstudio.com/.

Allaire, J.J. *TensorFlow for R*, https://tensorflow.rstudio.com/.

Bora, Neha. "Understanding Arima Models for Machine Learning." *Capital One*, https://www.capitalone.com/tech/machine-learning/understanding-arima-models/.

Feuerriegel, Stefan. *SentimentAnalysis Vignette,* https://cran.r-project.org/web/packages/SentimentAnalysis.

Hochreiter, Sepp. *LSTM Can Solve Hard Long Time Lag Problems - Neurips*. https://proceedings.neurips.cc/paper/1996/file/a4d2f0d23dcc84ce983ff9157f8b7f88-Paper.pdf.

Siami-Namini, Sima. *A Comparison of Arima and LSTM in Forecasting Time Series - NSF*. https://par.nsf.gov/servlets/purl/10186768.

Ulrich, Joshua. *Package 'Quantmod'*. https://cran.r-project.org/web//packages/quantmod/quantmod.pdf.

Wickham, Hadley. *Package "rvest",* https://cran.r-project.org/web/packages/rvest/rvest.pdf.