CSC 480 Final Project

Introduction

In the realm of stock market prediction, the primary goal is to predict the future value of a company's stock price. This goal is difficult to achieve due to the irrational nature and volatility of stock price data. Many studies have been done using more traditional methods with low levels of success, especially when creating longer-term predictions. In recent years, the advances in machine learning have made this goal much more attainable. With highly accurate classifications and a dependency on large amounts of data, machine learning is one of the most effective tools that can be applied to this problem. Existing literature often focuses on very short-term predictions – the next-day closing price of the stock is typically predicted in regression models, while many classification models predict whether the closing price will increase or decrease the following day. Although accurate and impactful, these predictions are limited by their narrow scope.

The aim of this project is to widen the scope to the medium-term, exploring the differences in accuracy when supervised machine learning models are trained to predict stock prices from one week all the way up to one month in the future. Predictions greater than one month into the future are deemed likely to be inaccurate given the highly erratic nature of the stock market. Regardless, if a long-term scope is even possible in this field, this project will aim to bridge the gap to it. The success of this project will have valuable impacts in the finance industry. Accurate short to medium-term stock price prediction will allow traders and investors to decrease their losses in short-term trading. There is the potential to lay down the basis for more successful strategies in personal finance, fund management, and portfolio management. Further to this, the exact nature of these predictions creates potential for them to be used in the development of automated trading bots in the future.

This project focuses on supervised regression models rather than classification. The set of algorithms used are multiple linear regression and the Long Short Term Memory (LSTM) Model, an advanced version of the Recurrent-Neural-Network (RNN). In existing studies, LSTM models often show very high accuracy when predicting day-to-day stock prices. The reason for this is that LSTM retains information from older stages in the neural network, in contrast to the more basic RNN. Although RNN may be accurate for short-term predictions, it is dependent on recent and current data because information from previous states does not persist. So, LSTM proves to be more useful in predicting medium-term stock prices. Meanwhile, the application of a multiple linear regression provides a second set of results with which to compare accuracy with the more advanced LSTM model. The significance of the difference in accuracies may provide insight into future application of machine learning to longer-term predictions.

Methodology

Stock price data is obtained from Yahoo Finance, with datasets containing the attributes of date, open, close, high and low. Yahoo Finance provides complete datasets, with data checks indicating zero missing values. The company's stock which will be used for this project is Apple (NASDAQ: AAPL). The data ranges over a ten year period from 26th November 2013 – 26th November 2023 for a total 2,532 trading days as sample size. Date and close are the primary attributes of interest, where close represents the stock price when the market closes on that day.

The data is first retrieved through the *yfinance* API in Python, which is then transformed into a DataFrame using the *Pandas* library in Python. With the closing price data, a number of further attributes are then extracted: momentum, volatility, and exponential moving average.

1. *Momentum*

$$Momentum = \begin{cases} 0, & close[i] < close[i-1] \\ 1, & close[i] \geq close[i-1] \end{cases}$$

2. *Volatility*

$$Volatility = \frac{close[i-1]\ -\ close[i]}{close[i-1]}$$

3. *Exponential Moving Average*
   The exponential moving average (EMA) is an advanced version of the moving average. The moving average calculates the average stock price over a given period of days. Meanwhile, the EMA applies higher weights to recent prices. This is useful given the fairly short-term nature of the predictions. For greater variety, the 7-day, 20-day and 50-day EMAs are tested as features.

$$EMA_{Today}\ =\ close[i] \times \frac{2}{1+n} + close[i-1] \times EMA_{Yesterday}$$

There are six total features: close price, momentum, volatility, 7-day EMA, 20-day EMA and 50-day EMA. After creating the data for these features, the first 50 days of data are removed from the dataset to ensure that the first row of data has a valid 50-day EMA, and by extension all other feature values are correct.

Predictions are then made for the stock price for the following intervals: one week, two weeks, and four weeks. The use of week-intervals as opposed to day- or month-intervals ensures that there is generally a future price to test the model on. This is important because the trading market is closed on weekends. For example, using Friday 1st September as a data point, the 4-week future price is on Friday 29th September, on which the trading market is open. In contrast, the one-month future price is on Sunday 1st October on which the market is closed, meaning that there would be no future data to train and test on. Using week-intervals generally fixes this issue, however, there is also the case of public holidays such as Thanksgiving on which the market is also closed. In these cases, the date is simply marked as null and dropped from the data before training and testing the models. For instance, with the example of Thanksgiving (23rd November 2023), the price data for 16th November is dropped from the training data for the 1-week predictions but not the other intervals, and similarly for 2- and 4-week prior data.

For both the multiple linear regression and LSTM model, a train-test split of 75-25 is used through the *scikit-learn* library. The regression model is trained and tested on week, 2-week and 4-week future prices, using residual sum of squares as the error. Coefficients are calculated for the previously mentioned features: price, momentum, volatility, 7EMA, 20EMA and 50EMA.

The LSTM model is implemented through the *keras* library. LSTMs are sensitive to the scale of the input data, so the data is first scaled used min-max normalization. The main input into the LSTM is the desired number of output units. This number is typically a multiple of 32 to assist with matrix multiplications within the network. For this project, a sequential model is made which involves stacking two LSTM layers on top of each other to improve the depth of the neural network, allowing the model to extract richer features. A dropout rate of 25% is set after each layer, meaning that 25% of the training data in each batch will be frozen. This prevents overfitting by ensuring the LSTMs are not overly fit to the data, and also speeds up the training time of the models. At the bottom of the stack, two core dense layers are added. These layers reduce the output size from large numbers to single output of 1. The exact output number for each of these layers in the overall sequential model is determined in testing using hyperparameter tuning.

Other parameters which are tuned for the LSTM model include the optimizer and the loss function. In this case, Adaptive Moment Estimation ('Adam') is chosen as the optimizer because it is generally regarded as a robust choice for neural networks, while the loss function is decided between mean-squared error (MSE) and mean absolute error (MAE). MSE penalizes large errors more than smaller ones compared to MAE.

Model performance is evaluated through mean-squared error (MSE) and the coefficient of determination (R^2). R^2 provides a good reference point for how much of the differences in MSE is due to data variance between the respective models.

$$R^2 = 1 - \frac{MSE}{Var(y)}$$

Results

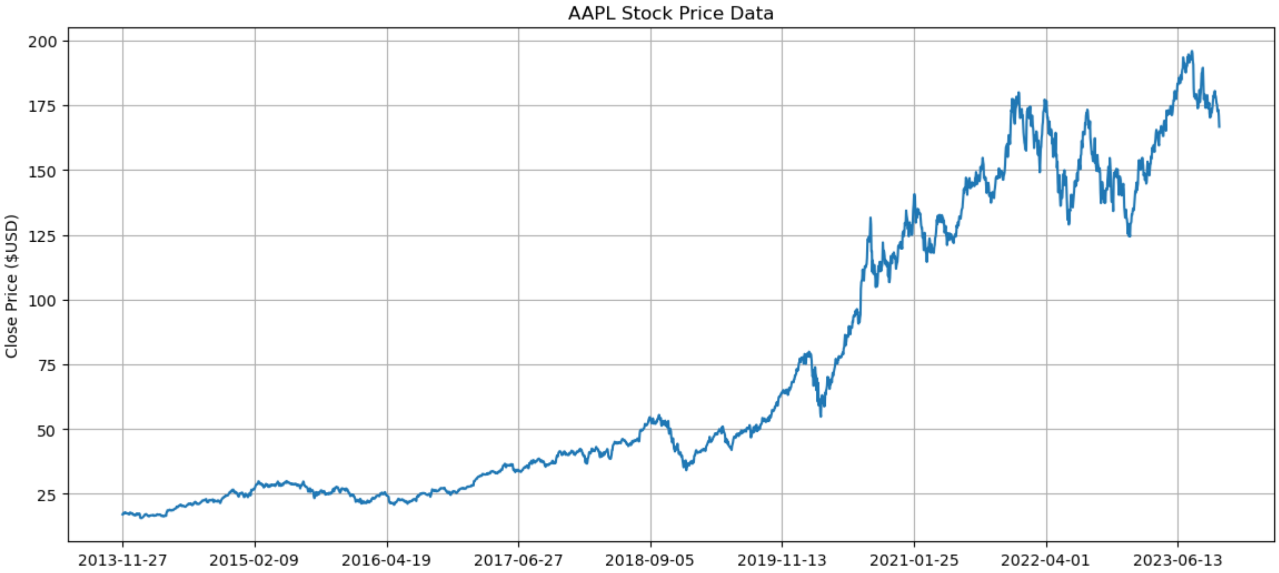

Figure 1: Apple 10-year stock price data retrieved from *Yahoo Finance*

These first figures plot the data before any model training or testing. Fig. 1 plots the AAPL stock price over a period of 10 years. Fig. 2 and Fig. 3 shown below only include data in the one year period from Nov 2022 – Nov 2023 for better readability.

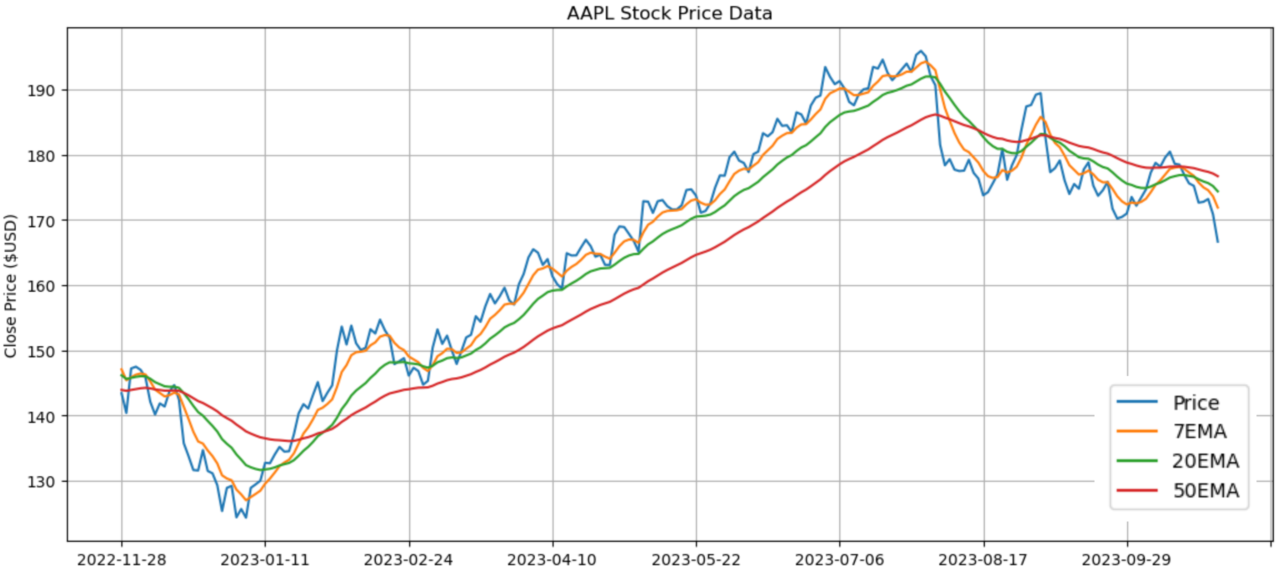

Figure 2: Apple 1-year stock price and exponential moving averages (7, 20, and 50-day)

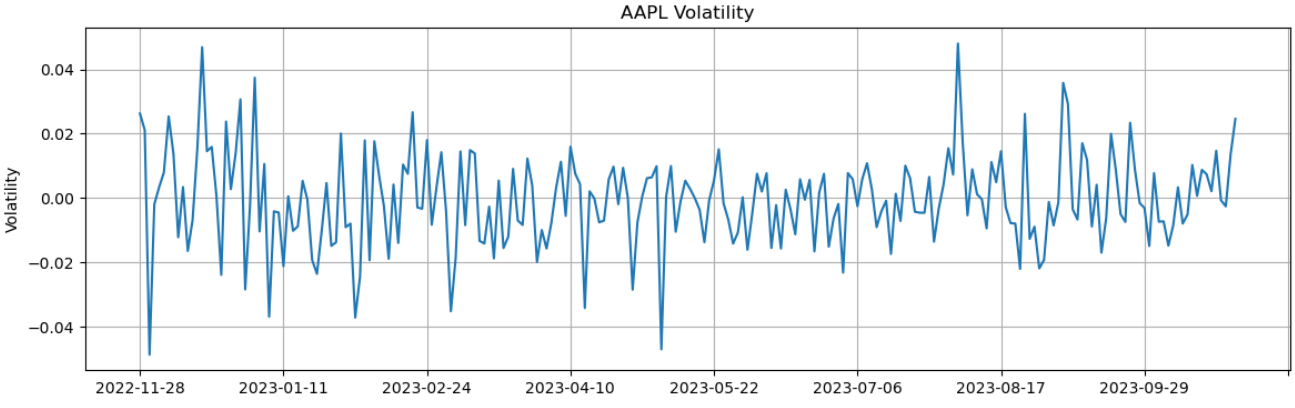

Figure 3: Apple 1-year stock price volatility

Fig. 4, Fig. 5 and Fig. 6 are plots of the predicted 1-week, 2-week and 4-week future prices as predicted by the multiple linear regression model. As expected, the longer the interval of prediction, the less precise the prediction appears to be.

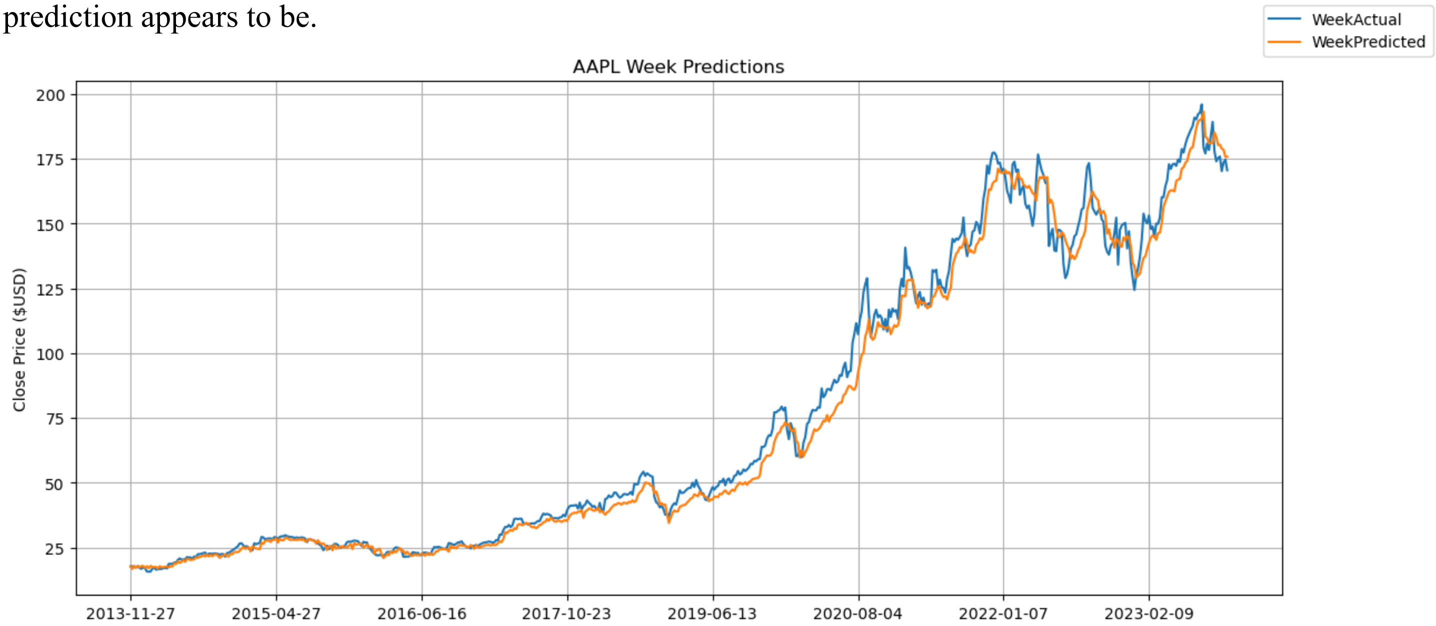

Figure 4: 1-week predictions for Apple stock price

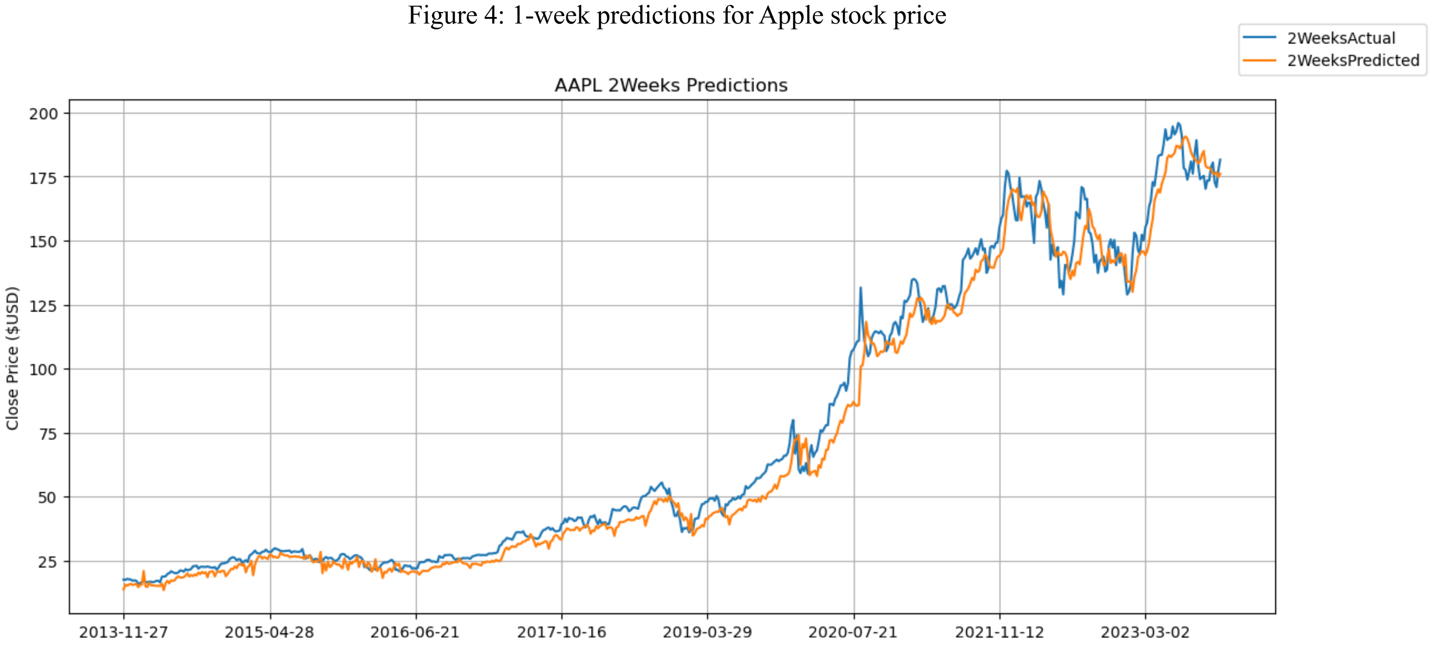

Figure 5: 2-week predictions for Apple stock price

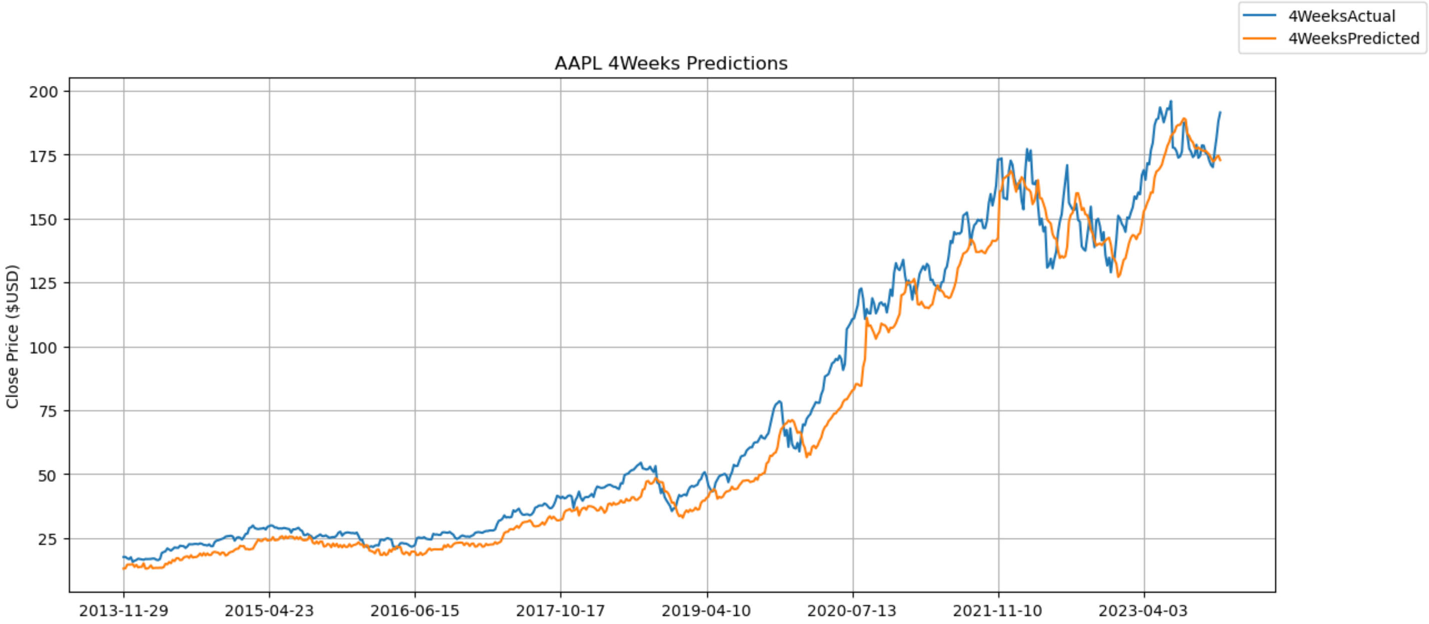

Figure 6: 4-week predictions for Apple stock price

Moving on to the LSTM model, hyperparameter tuning led to the development of the final sequential model. The two LSTM layers had output values of 1024 and 64 respectively, followed by the two dense layers with output values of 32 and 1 respectively. The optimal loss function was found to be mean absolute error, which indicates that penalizing large errors less resulted in improved overall performance of the model. A comparison of the final LSTM model performance with the multiple linear regression model is outlined in Fig. 7, with MSE and R^2 as metrics.

Model	Multiple Linear Regression		LSTM	
Metric	MSE	R^2	MSE	R^2
1-Week	33.8880	0.9892	14.2482	0.9954
2-Weeks	52.2637	0.9832	21.4378	0.9927
4-Weeks	101.4898	0.9680	51.6228	0.9827

Figure 7: Performance metrics for multiple linear regression and LSTM

As a result, Fig. 7 reveals that the LSTM performed better in all categories when predicting future stock prices. The MSE increased and the R^2 decreased as the time interval was larger, which is expected. Both models had a very accurate R score, indicating the exactness with which future stock prices can be predicted using machine learning algorithms.

In the future, this project can be improved by testing these models on further datasets rather than just the one Apple dataset, such as an index to provide a better estimate of the overall direction of the market. Furthermore, the performance of LSTM can also be compared with other machine learning algorithms such as Random Forest and Support Vector Machines. Further features can also be added to improve the dimensionality of the models.