Elvis Atitsoabui

**Date:** 03 May 2024

***Analyzing and Predicting Stock Market Behavior Using Machine Learning***

**I: Introduction**

**Motivation**

The main goal of our project was to understand stocks and their behavior and especially their volatility. We wanted to do this using machine learning methods and time series techniques that were adapted to work with time dependent data. From this, we hoped to be able to make accurate predictions on stock price. The motivation for this project came essentially from the core of investment analysis. With the rise of artificial intelligence, we see that many traders are relying on algorithmic trading to predict and gain information on stocks. Not just this, but new methods, models, and techniques are being derived to tell traders and investors what stocks to trade and when. With the help of new effective machine learning techniques, prediction on volatile time dependent data such as stocks has become much easier and much more predictable then it was even 10 years ago. Our project was directed at exploring how powerful these techniques are, what the conditions of their effectiveness are, and just exactly how much these techniques could do. Our resulting models serve as a first step in trying to minimize volatility and maximize return on investment.

**The Dataset**

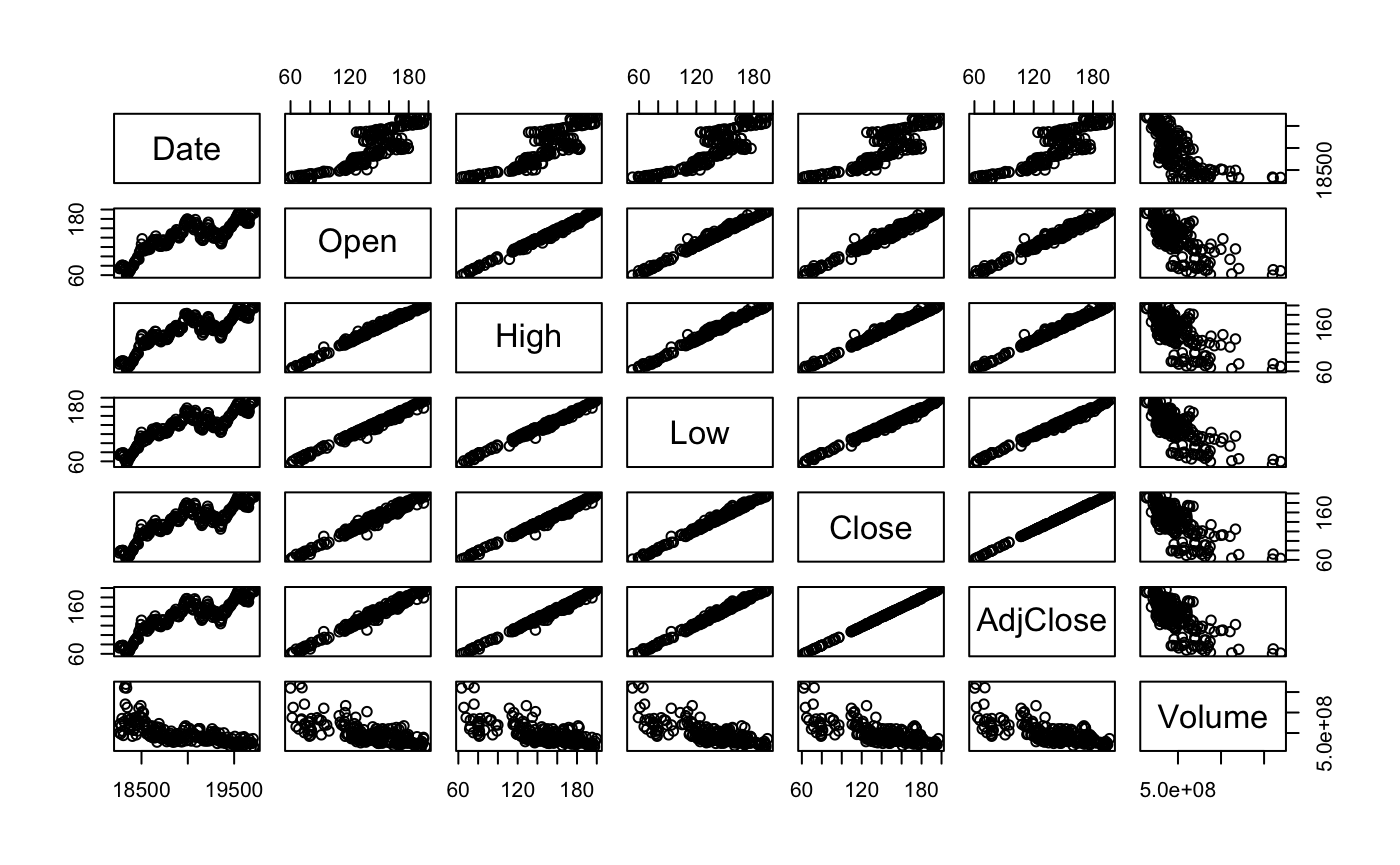
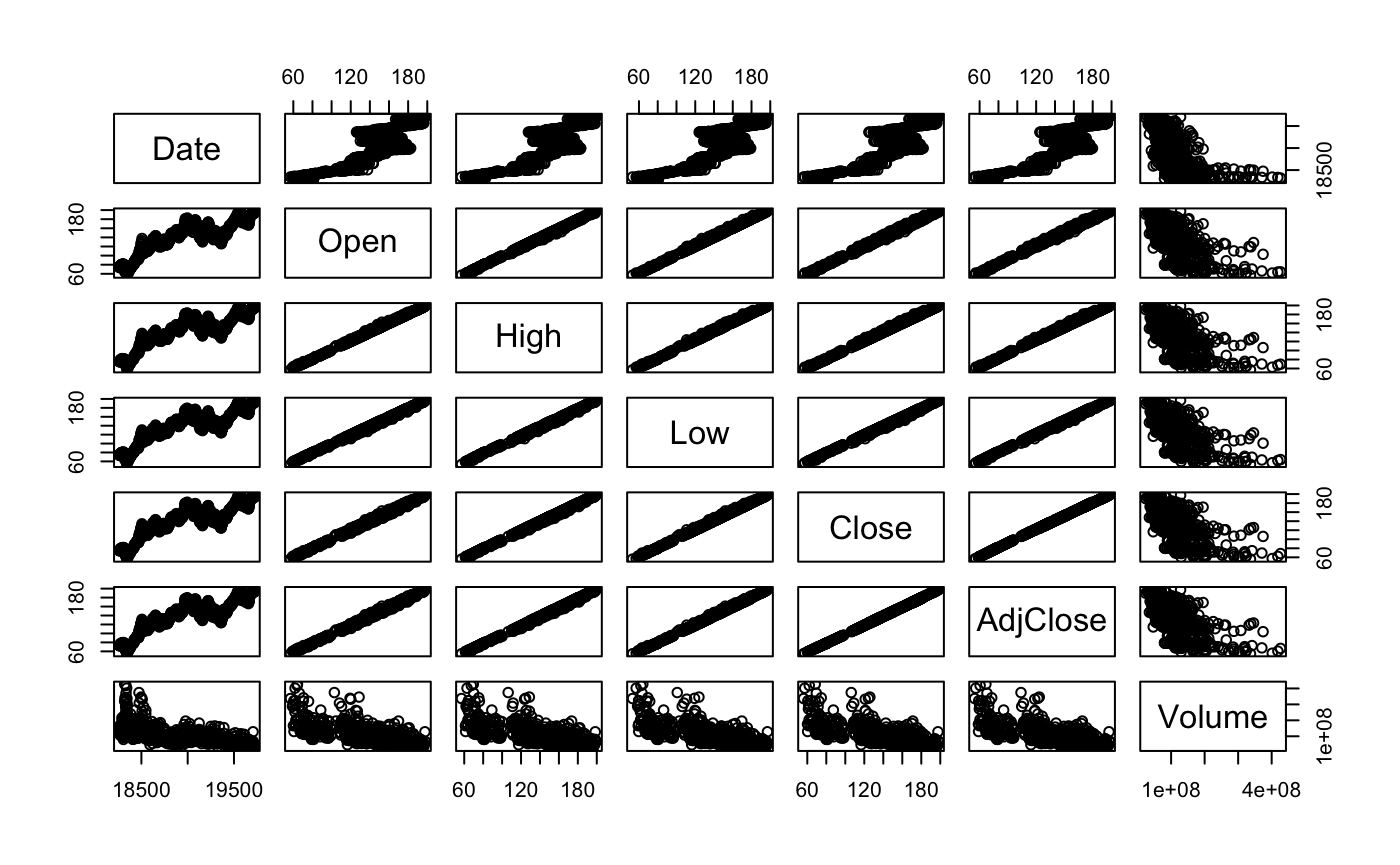
The datasets used for this analysis were taken from Yahoo Finance. It was the Apple (AAPL) stock data from over a four-year timespan (January 2020 - December 2023) and consisted of three separate datasets broken down by timeframe: Daily, Weekly, and Monthly. This way, we could measure the effects of our various methods of predicting average return on different time scales. Each dataset recorded six numerical variables in addition to the date. ‘Open’ was the price of the stock at the beginning of the stock market (Monday 9:30 AM, EDT). ‘Close’ was the value of the stock at market close (same day at 4:00 PM, EDT). Of course, weekends were excluded from the dataframe since the market is closed. The ‘High’ and ‘Low’ were the maximum and the minimum of the stock prices, respectively, over the given timeframe. ‘Volume’ represented the amount of trades made in that same timeframe. Lastly, ‘Adjusted Close’ was the closing price of the stock adjusted for dividends and the amount needed to pay back shareholders.

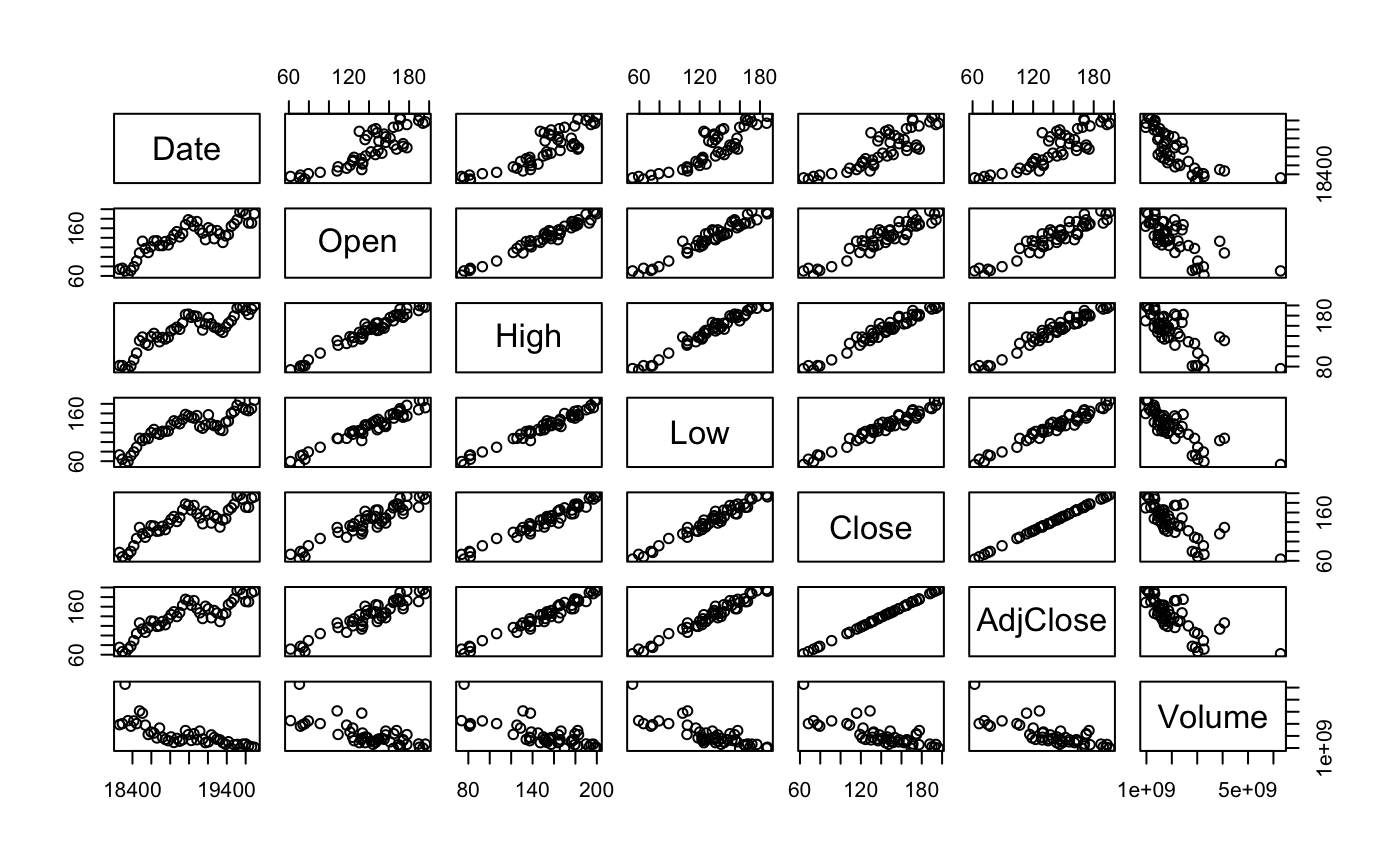
Adding to this dataset, we created an additional column designated for ‘Returns’. We calculated these by subtracting opening stock prices from closing ones, then dividing by those closing values. This reflected the change in the stock for one time-step of its respective timeframe. Should the stock price have gone up in one time-step, then the value would have been positive; had it gone down, the value would have been negative. Understanding the closing values and their role in returns was as close to predicting the stock in real time that we could get without using actual tick data, which is more comprehensive but significantly more expensive and computationally complicated.

**Exploratory Data Analysis**

Since little preprocessing was required given our Daily, Weekly, and Monthly datasets, the next step was to perform an exploratory analysis to pickup on trends in the data and get a better feel for variable interactions and behaviors. The goals of this initial analysis included gathering summary statistics, observing pairwise plots and correlation matrices, investigating multicollinearity, calculating and visualizing distributions of returns, and observing distribution of differences between variables. This last goal was added since variable differences such as those used directly in the computation of returns could tell us more about these relationships.

The first notable finding was seen in the three pairwise plots. The figures below advance from Daily to Weekly to Monthly pairwise plots for the seven variables in the datasets. We see that variability decreased as more data points were added to the set. It was also easy to see high, strong, positive correlations between some of these variables.





Regarding the investigation on multicollinearity, we decided a simple observation of correlation matrices and these pairwise plots was not enough, so we checked the proportion of variance explained by eigenvalues of variables, which is seen below for the Daily, Weekly, and Monthly datasets.



We also calculated condition numbers which told us, along with the values above, that we would need to address the issue of multicollinearity in each of the respective models and we would have to proceed with caution. For the remaining goals, the summary statistics were as expected, we found that return distributions are roughly Normal, and we concluded that features of differences in variables had similar results as those from our multicollinearity investigation.

After this initial analysis, we were ready to proceed with our predictive models. Based on our findings, we had two time series techniques for stochastic modeling, which included an Autoregressive (AR) model and an Autoregressive Integrated Moving Average (ARIMA) model. Next, we explored Long Short-Term Memory (LSTM) Neural Networks which are well-suited for time series data since forget gates and memory cells help capture patterns better than for regular neural networks. Lastly, we explored tree models since they are common in the industry, robust, and interpretable. They were great for picking up nonlinear complex relationships and have built-in functions to identify feature importance which could be useful.

**II: Methodology**

**Long Short-Term Memory Neural Networks**

***Why LSTM?***

First, we decided to fit the LSTM model to our due to its ability to pick up on long-term time dependencies. These very crucial facts are needed in determining and predicting stock prices because of the huge amount of volatility involved. The ability of the LSTM to pick up on these time dependencies is due to the fact that it uses various gates to regulate and delete information. This helps to stop the vanishing gradient problem of the RNN and keep accurate predictions and make sure the model does not overtrain. This feature makes the LSTM very adept at understanding complex long term time dependencies.

What we were expecting from our results of fitting the LSTM was the best fit of the data compared to the previous models. We had expected that the LSTM would yield the lowest RMSE and produce visuals that were extremely close to the actual true closing and average return values that we were predicting.

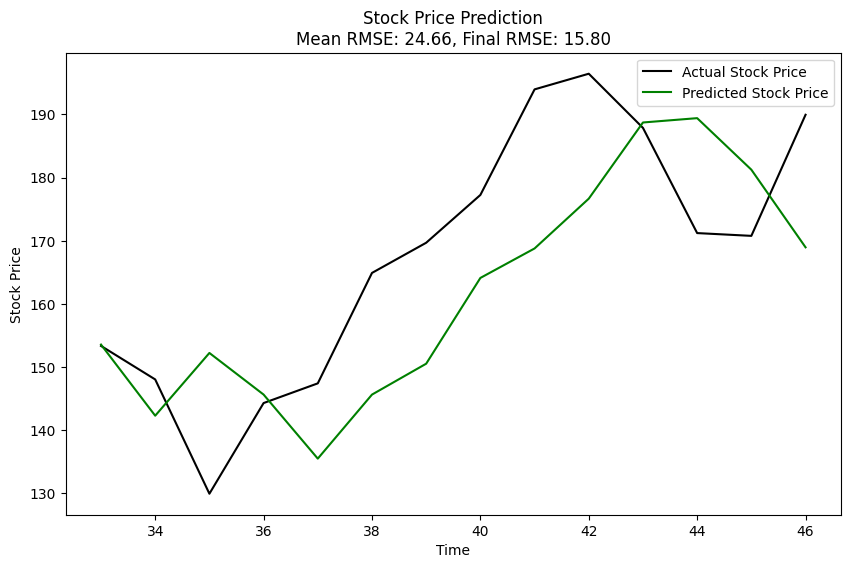
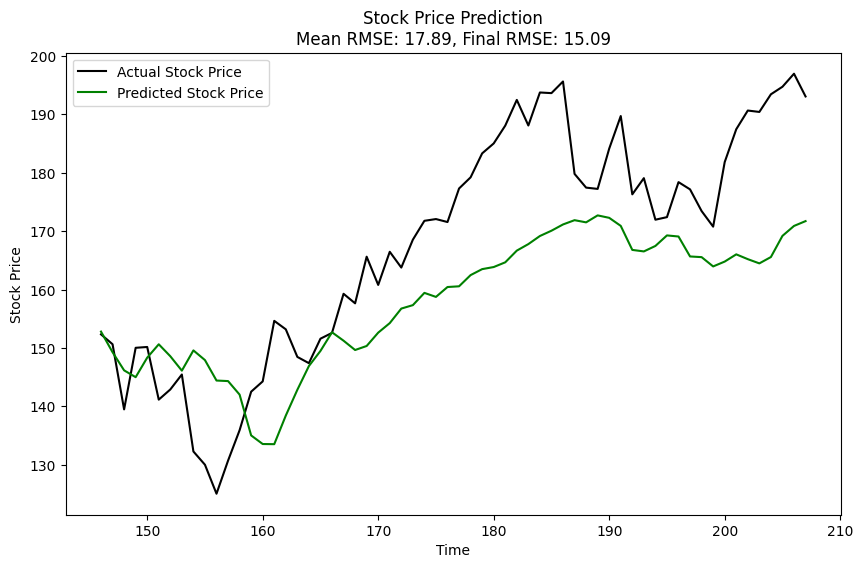
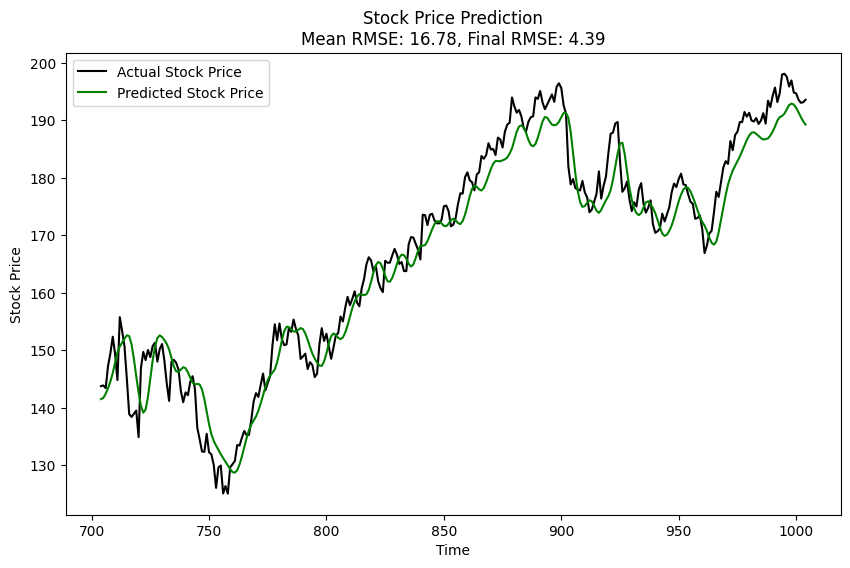
***Results***

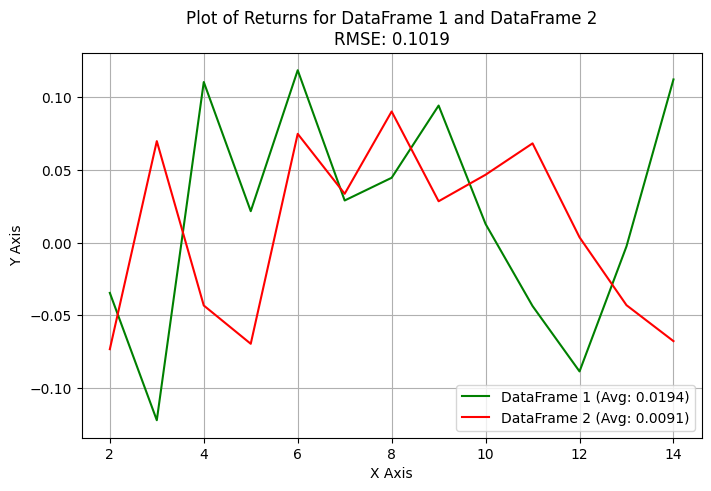
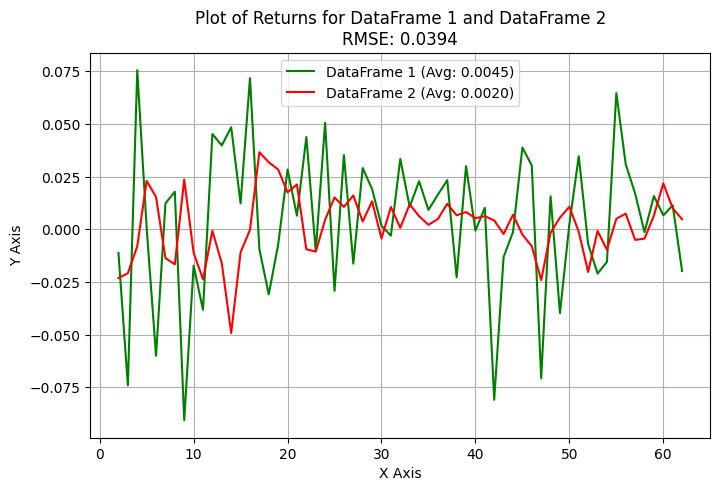
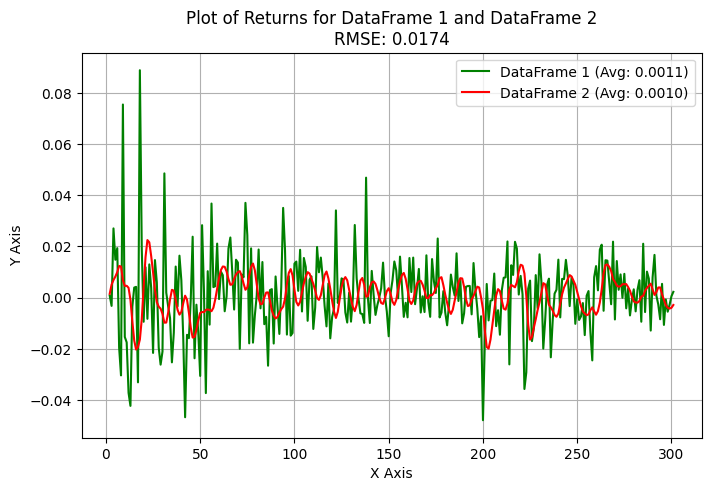
For the Daily dataset, we obtained an RMSE of 4.39 on the prediction of our closing price and a return rate prediction of 0.10 when the actual rate was approximately 0.11. These results were the best for the LSTM model in terms of the other datasets. This makes sense given that the Daily set had 704 observations giving the model more to learn from. After trying multiple parameters, the one that gave us the best results as described were 4 hidden layers, 50 units per layer, 50 time-steps, and 100 epochs.

For the Weekly dataset, the RMSE for the closing prices shot up to 15.09 and the RMSE for the average return rate was 0.0394. The actual rate was 0.0045 while the predicted was 0.0020. The increase in error is due to the fact that we had a decrease in available observations for the model to train on, so the test set had a decrease in accuracy. Despite this, our model was able to follow the general direction of the stock, which is still very valuable as predicting points of increase and decrease can aid in stock prediction and investment. The parameters for the Weekly that gave the best predictions were 5 hidden layers, 50 units per layer, 50 time-steps, and 100 epochs.

Our results for the Monthly dataset were similar to the Weekly results. The RMSE for closing was 15.8 and the RMSE for the return was 0.1019. The predicted average return was 0.0091 and the actual was 0.0194. Again, this increase in error is attributed to the fact that we had less observations in our training set than the Daily. Still, the ability of the model to predict general uptrends and downtrends of the stock is still very beneficial to traders and investors when trying to determine an optimal investment method. This model was trained using 5 hidden layers, 50 units per layer, and 2 time-steps.

The visuals below provide a better understanding of the actual trend of the stock and our predictions. DataFrame 1 represents the actual return rate and DataFrame 2 represents the predicted return rate. The graphs go from Daily to Weekly to Monthly as you advance from left to right.





**III: Conclusion**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Daily RMSE** | **Weekly RMSE** | **Monthly RMSE** |
| **AR** | 0.0165 | 0.0633 | 0.1028 |
| **ARIMA** | 0.0211 | 0.0420 | 0.0922 |
| **LSTM** | 0.0174 | 0.0394 | 0.1019 |
| **Tree Model** | 0.0239 | 0.0407 | 0.0888 |

In general, Daily data proved to be the best for our models. This makes sense as it is the most collected data, and the time between each object allows for more subtle variations. The best RMSE was achieved by the Long Short-Term Memory (LSTM) model for Daily data. However, the Autoregressive Integrated Moving Average (ARIMA) and Tree models were close competitors and could be used interchangeably.

In conclusion, if you aim to predict returns for stock data, any of these three models could be applied to your daily data. The choice depends on the complexity of the model you wish to create, with LSTM being the most complex and the Tree model being the least complex. This decision should be based upon your real-life constraints.

Predicting returns, or any other value, is beneficial because the better you can predict stock behavior, the more money you can potentially make in the stock market. However, it’s important to remember that while these models can provide valuable insights, they should not be used as the sole decision-making tool for stock trading due to the inherent risks and uncertainties in the stock market.