CS 4641-A

Project Midpoint Check In

Submitted to:

Dr. Madhi Roozbahani

Date: November 17, 2021

Submitted by Team 13:

Ethan Benville, Carson Earnest, Alex Reyna, Jim Sherman, Conor Walsh

Introduction

The S&P 500 stock index consists of many of the top global companies in terms of market cap across many sectors. To assess a company's financial performance, shareholders and investors utilize a wide variety of metrics to predict how market factors (ex. changes in supply chain demand, industry competitor sales growth) or business developments (ex. new product launch, company merger) may impact a company's stock price on a given day. As these prices can fluctuate greatly on an inter-day and intra-day basis, machine learning algorithms can help create models which predict daily stock price changes and give insight into how historical trends may influence future stock price volatility.

Problem Definition

The main problem we are seeking to solve with this project is finding a more accurate method of predicting future stock prices using machine learning algorithms. Originally, we aimed to do this by feeding the algorithm historical information about a company's fundamentals, meaning its core business and valuation. These would be metrics such as revenue, margins, the stock's beta, and the stock's price-to-earnings ratio. However, due to the difficulty of finding historical fundamental information on a day-by-day basis, combined with the fact that some of these metrics remain static for long periods of time (revenue and earnings are reported quarterly), we decided to change our approach to price-action trading. This is a process by which traders analyze the stock price exclusively with the intention of predicting future returns. New features may be created based on the price, such as a moving average, but the analysis is still based solely on the stock's price at its core. We anticipate that the algorithm will likely be more accurate in predicting shorter-term yield compared to long-term yield, as a company's long-term performance is typically based on business fundamentals and earnings, which our algorithm does not take into account.

Data Collection

In order to pull the necessary data for testing and training, we used the yahoo finance API. We pulled two primary data sets: the first was daily price data from September 9th, 2010 (oldest data we could pull) until December 31st, 2020, and current financial metrics about the companies in our testing pool. For the midterm report, we broke up the training data into training

and testing data. However, for the final report, we will pull daily pricing data from January 1st, 2021 until the present for the testing data.

Once the data was collected, we performed two methods of preprocessing. The first was to create 10, 50, and 200-day moving averages as well as 5, 20 and 260 day forward returns for the prices. It is important to note that the 5 day period is equivalent to a week of trading days, 20 days is equivalent to a month of trading days, and 260 is the actual number of trading days in a year. The moving average was calculated by pulling the average price over the specified period. The forward return was calculated with the following formula: (price of day i+[4] - price of day i)/(price of day i) for 5 day forward return, (price of day i+[19] - price of day i)/(price of day i) for 10 day forward return, and (price of day i+[259] - price of day i)/(price of day i) for the 260 day forward return, where the range of the indexing variable i started at 1.

Methods

For this part of the project, we implemented ridge regression in python and divided our data into training and validation/testing data. Sklearn was used to split the data, create a ridge regression model, and evaluate error measurements. Data was extracted from our cleaned csv file for the ETF VOO to use in our examples, however, any of the tickers can be used. All moving averages including 10 dma, 50 dma, and 200 dma were then extracted from the resulting data frame. Next, the data was split into 80 percent training and 20 percent testing, Next, ridge regression is applied, and the number of days to predict in the future are specified (in our case, 30 days), and matplotlib is used to create graphs, including forecasts, ridge regression prediction values, and the actual data. Error measurements (RMSE and R squared) are calculated to evaluate the effectiveness of the model.

Results / Discussion

Examples of the moving averages plotted with their respective stock prices are shown in figure 1. Notice that the 10 day moving average is the closest to the actual trend in the data. This makes sense, as the 10 day moving average is more representative of a current day's stock price compared to a 50 or 200 day moving average.

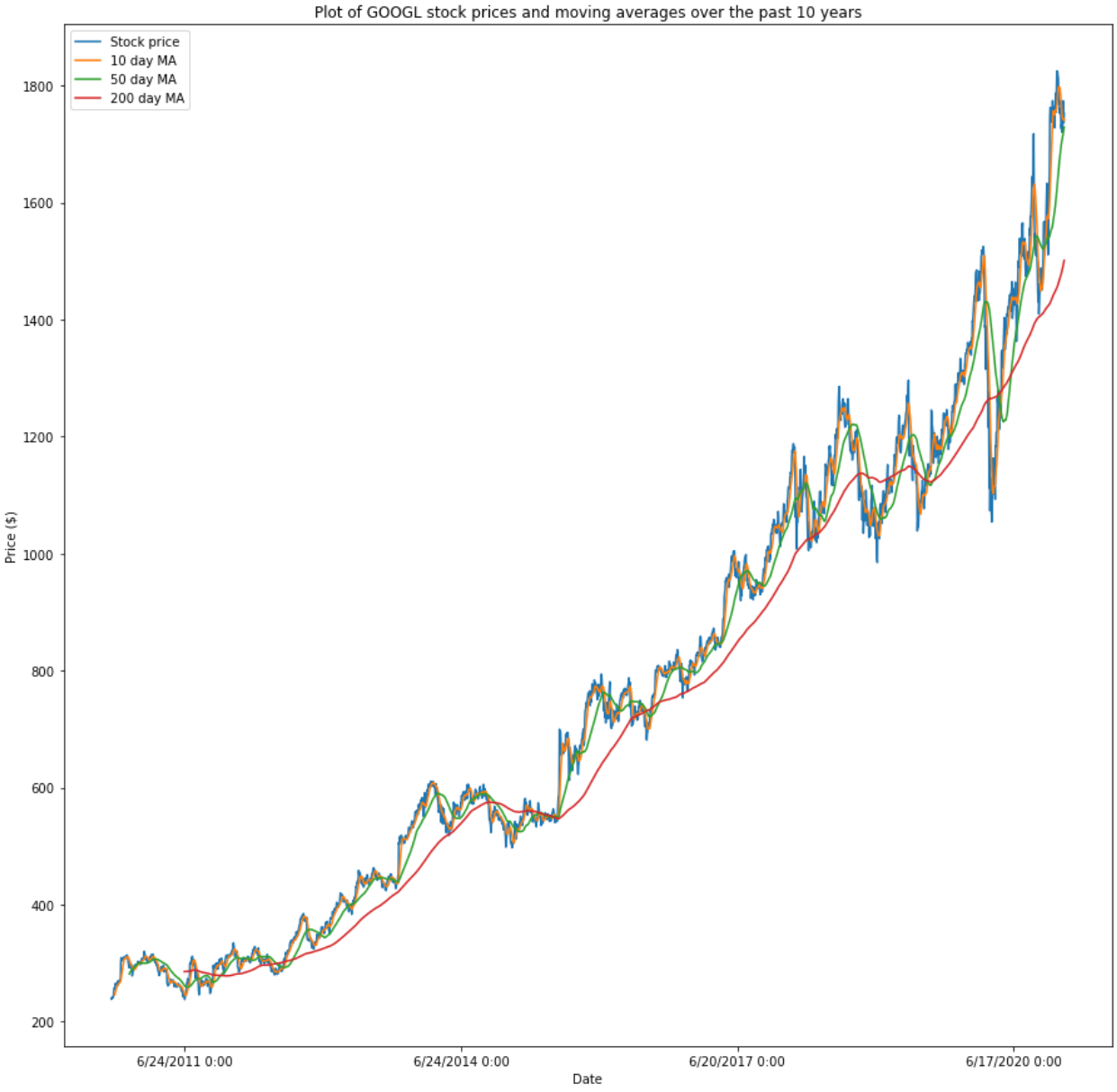

Figure 1: Plot of stock price by day along with moving averages

Supervised Learning: Ridge Regression

A train test split was performed with a test size of 0.2. We used R-squared and RMSE to evaluate our models. The results suggest that predictions longer into the future tend to be less accurate. Examples of the ridge regression model using the moving averages as features are shown below in figure 2. Each ridge regression graph is accompanied by another graph (shown

in figure 3) that displays the difference between the predicted stock price (red dots) and the actual stock price (blue line). Each graph has the respective RMSE and R-squared value generated from applying ridge regression to the data. The RMSE value expresses how well the predicted stock prices created from the training data match the actual stock prices. The R-squared value expresses how well the test data of the different moving averages predicts the actual test data of the stock prices.

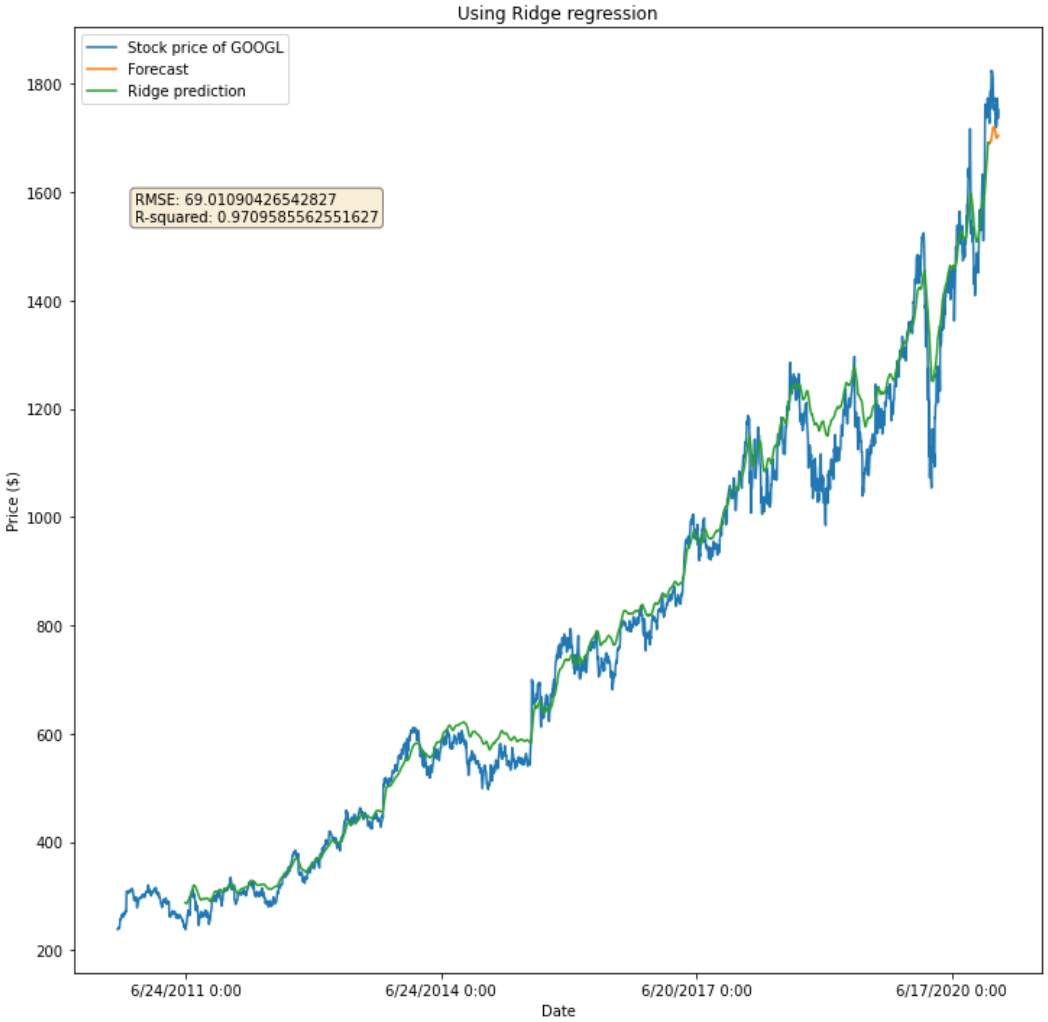

Figure 2: Fitted ridge regression stock price values

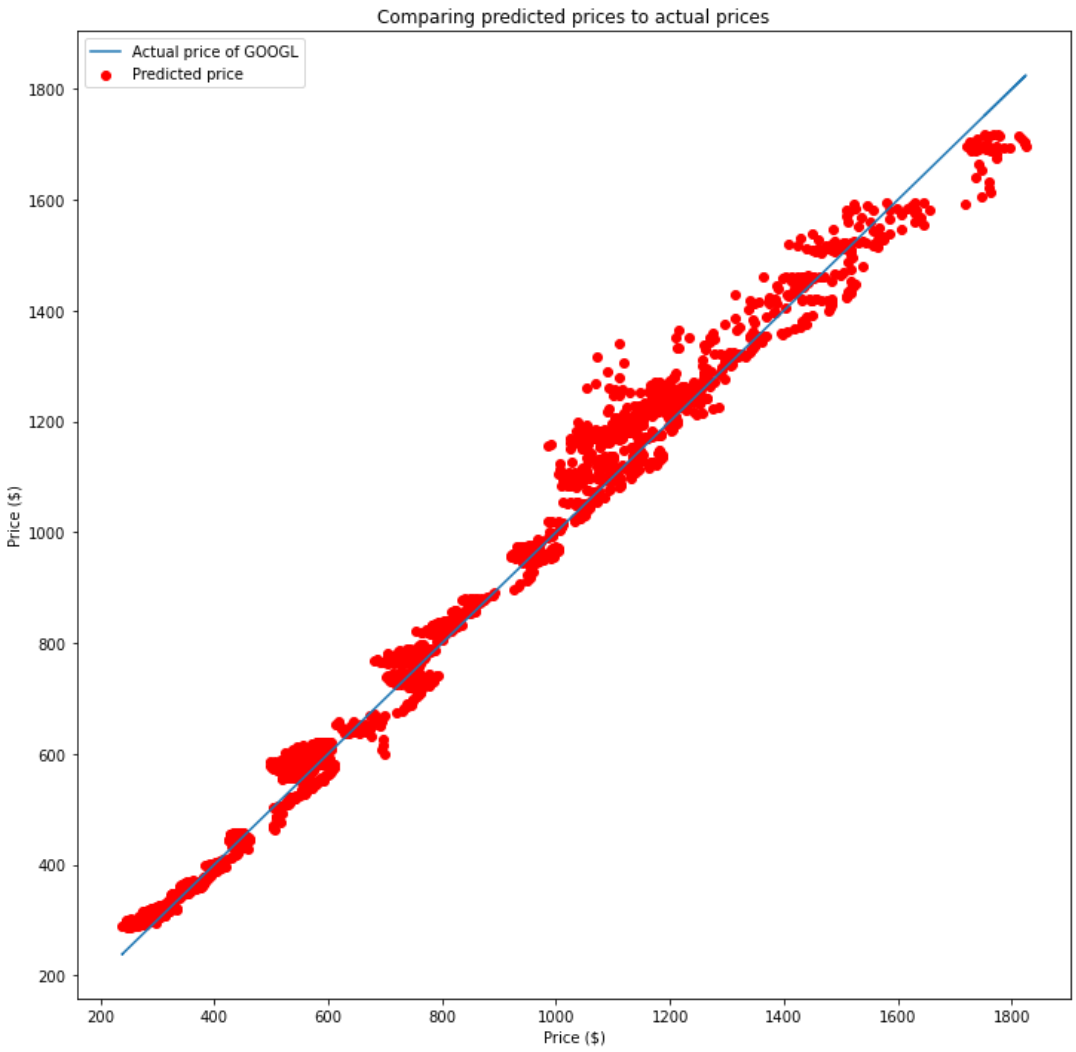

Figure 3: Plot of ridge regression stock price predictions vs actual stock prices

Notice the RMSE values are not very close to 0. However, compared to the 2500+ data points we used, this RMSE value is relatively low. We can tell this value is relatively low by looking at how far off the red dots are from the blue line. For each of the examples above, the predicted prices (red dots) are relatively close to the actual price (blue line). As for the R-squared value, for both graphs, this value is very close to 1, which means the test data predicts the price of the stock pretty accurately. However, both the RMSE and R-squared values may increase or

decrease depending on how much data we use to train and how much we use to test. Below in figures 4, 5 and 6 are examples of one stock's regression model using testing data of 25 days, 50 days, and 100 days.

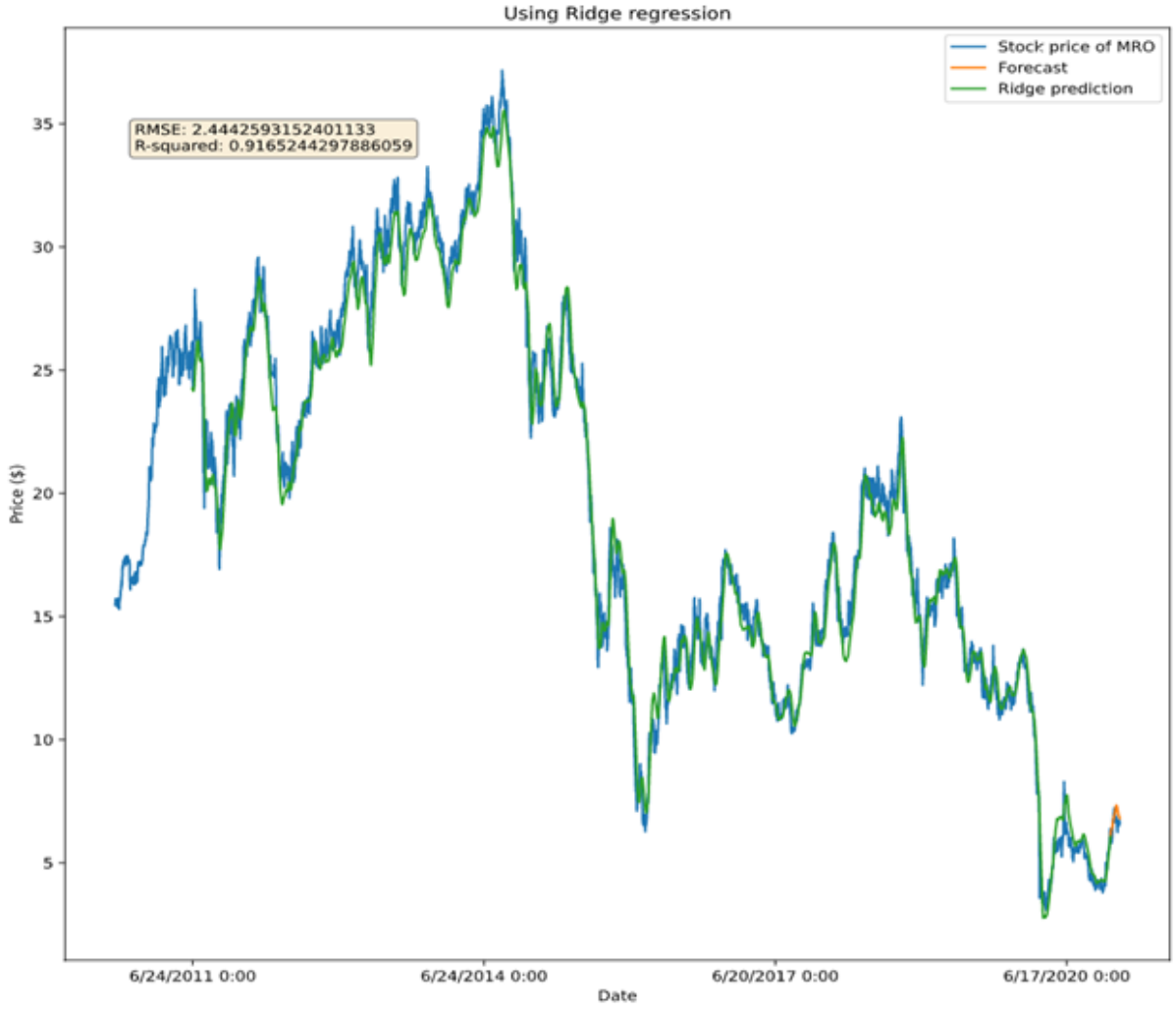

Figure 4: 25 day forecast using ridge regression stock price predictions

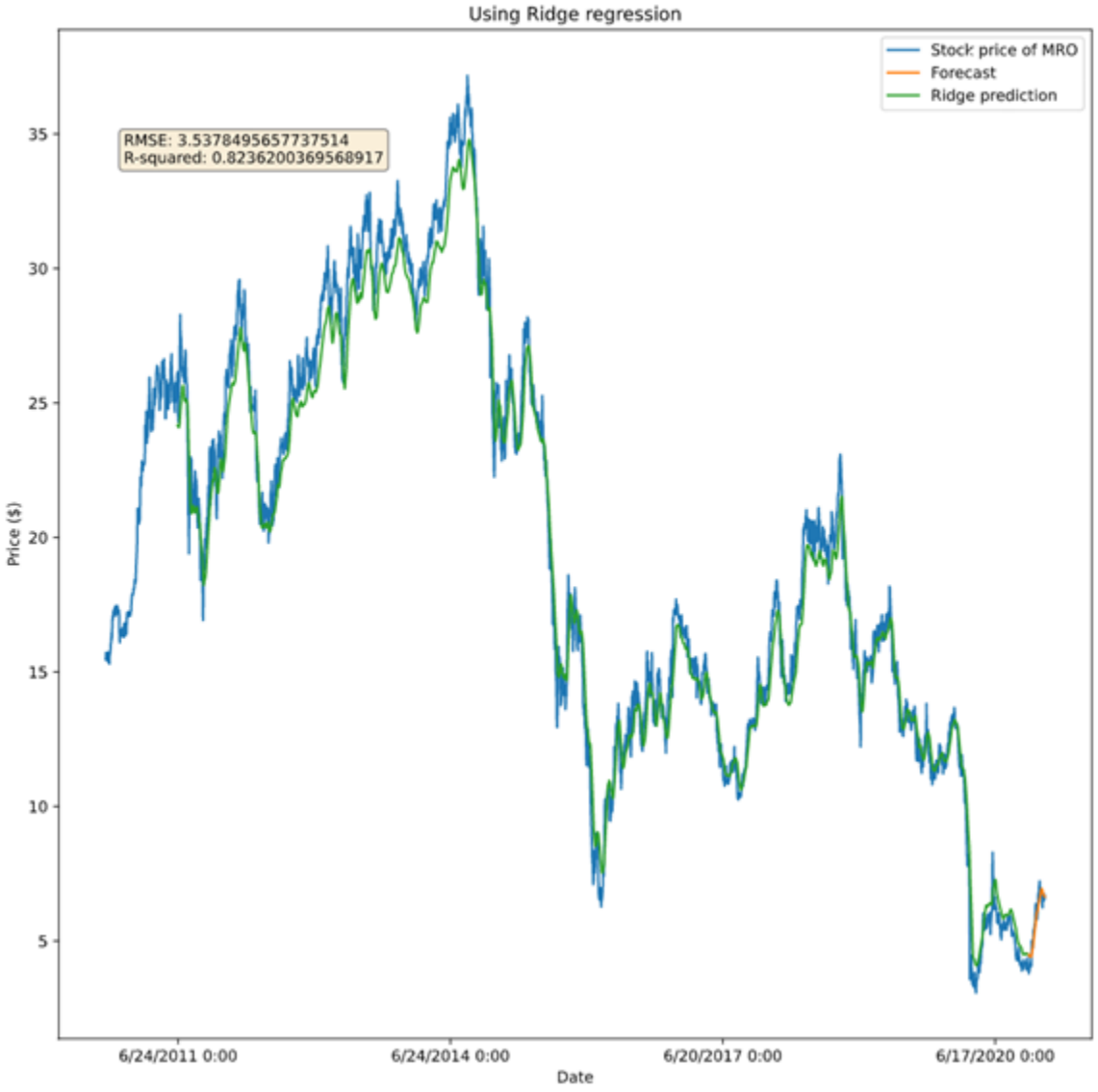

Figure 5: 50 day forecast using ridge regression stock price predictions

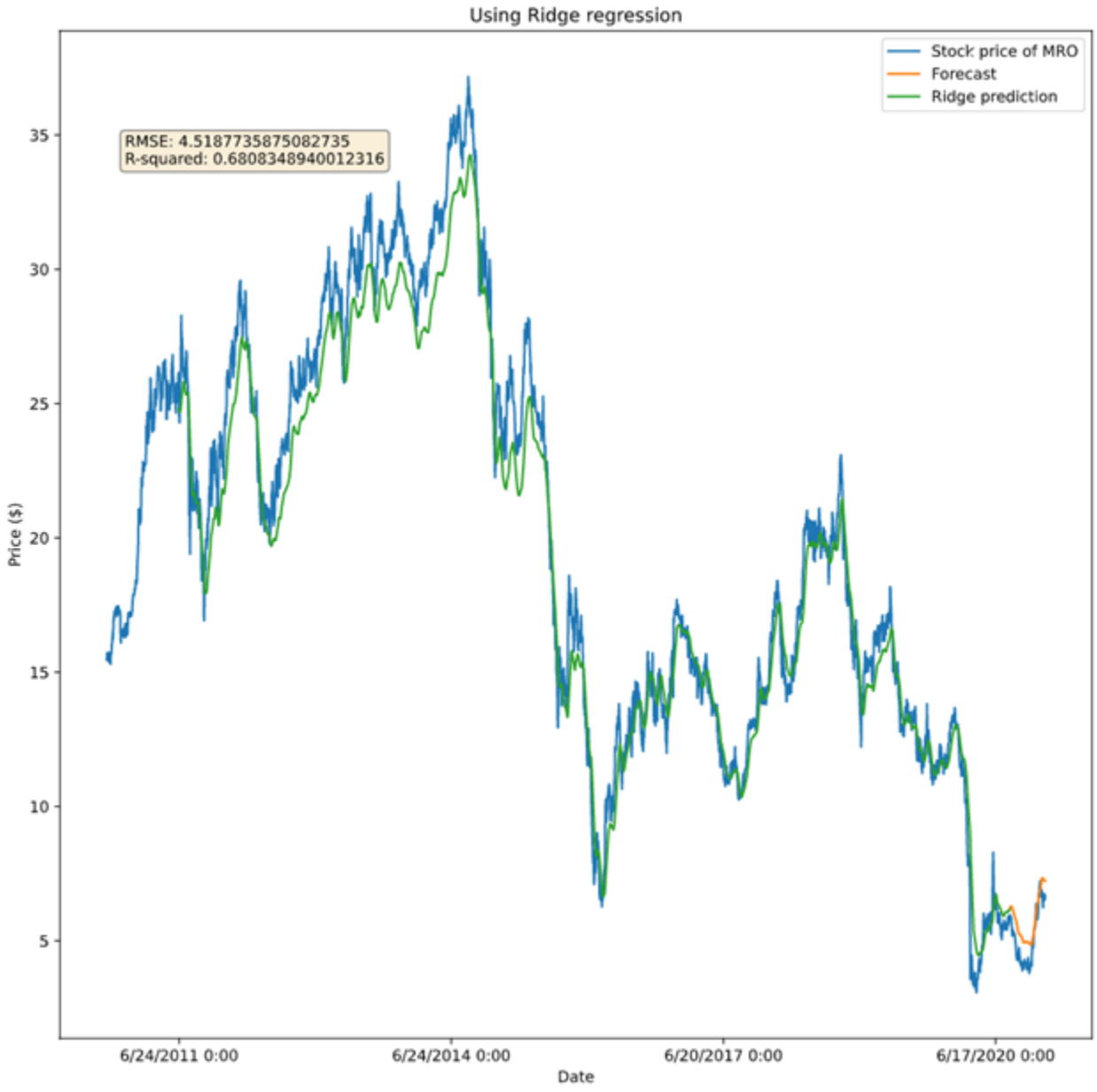

Figure 6: 100 day forecast using ridge regression stock price predictions

Notice that as we increase the number of days we are predicting, which decreases the size of our training, the RMSE values increase and the R-squared values decrease. These trends in RMSE and R-squared combined mean that the model gets less accurate the further you try to

predict in the future. Therefore, ridge regression is best applied to stock price determination if you are only determining a short number of days ahead. This follows what we predicted earlier that ridge regression is more accurate for short term prediction compared to longer term prediction.