Baseline Stock Prediction: Fundamentals of Statistical Machine Learning

Evan Abbott
University of Tennessee
Knoxville, TN
eabbott9@vols.utk.edu

Isaac Channing
University of Tennessee
Knoxville, TN
ichannin@vols.utk.edu

Beatrice Eldridge
University of Tennessee
Knoxville, TN
beldrid4@vols.utk.edu

I. Introduction

According to a 2023 Gallup poll, 39% of Americans own no stock, mostly because they're too scared to put money in [1]. After all, the stock market is hard to predict, and most people don't have the time to try to sift through news and stock prices to make the best decisions possible. With a stock predictor, Americans would be less scared to invest, helping them develop their portfolio earlier in life, thus benefiting their retirement. By training on Berkshire Hathaway stock data from 2015-present, we developed a linear regression-based stock predicting model that does just that.

In this project, we'd like to cure the stock CSV file, create a linear regression model that predicts future Berkshire Hathaway stock prices with an MSE less than 3000, and compare the linear regression model to a Lasso regularization model with performance metrics like MSE and R2 score. Also, later in the semester, we would like to implement a more complex polynomial regression model to hopefully improve performance.

This paper will discuss the Berkshire Hathaway Stock Price Data dataset and how we cured the data to prepare it for the linear regression model. Then, we'll discuss why we chose the linear regression model and analyze its performance in this dataset. We'll also compare the regression to a Lasso Regularization model to see which is more accurate. Lastly, we'll propose an extension to our project, the polynomial regression model, to hopefully improve performance.

II. Dataset

A. What the Dataset Covers

Berkshire Hathaway is a holding company, meaning it does not provide goods or services to the public. Instead, the organization invests in other companies as a middleman for other investors; investors can avoid risk by buying stocks here instead of investing directly into several companies. Our dataset (as seen in Fig. 1) merely tracks the stock price of Berkshire Hathaway from 2015 to 2024. From a quick glance, one can identify the relatively consistent upward trend of the stock price.

B. Data Curation Techniques

This dataset has several statistics that describe the stock price for a given day. Our data curation only needs a singular price value per day, which is the average of the open and close values for any day. If the open or close value was used instead, then the data would be more subject to random fluctuations in the true data; the average softens this noise.

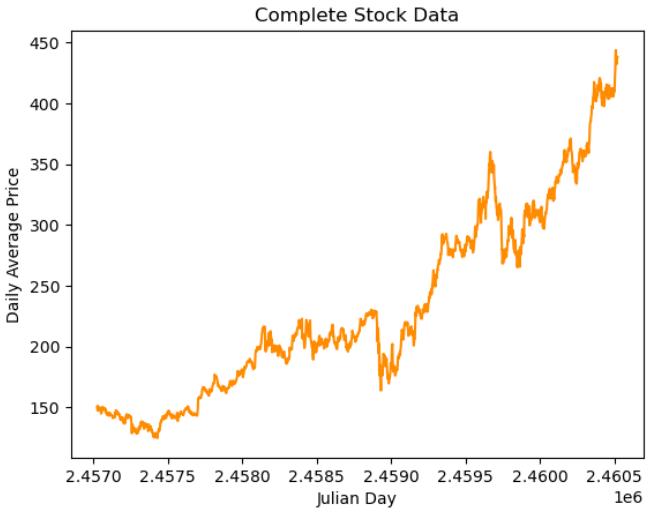

Fig. 1. The dataset before the train and test split.

Another step in the data curation was converting the dates, the x-axis, to their corresponding Julian days. In astronomy, a Julian day is the number of days since a certain point: 4713 BC, the first Julian period. This conversion converts dates that are string data types into precise integers, which allows for easier interpretations of graphs.

Finally, a standard scaler is used to linearly transform the data such that the average stock price value is zero with a standard deviation of one. This feature scaling reduces the total range of the x-axis values and will prepare the data for any algorithms that might expect a standardized dataset.

Beyond these three forms of data curation, we also split the data between training and testing sections. Because our linear regression model is only used to predict future stock prices, the first eighty percent (80%) of the overall data was used to train our linear regression model. This split can be seen in Fig. 2.

After all of our data curation, our final dataset contains a list of Julian days and the corresponding average of their open

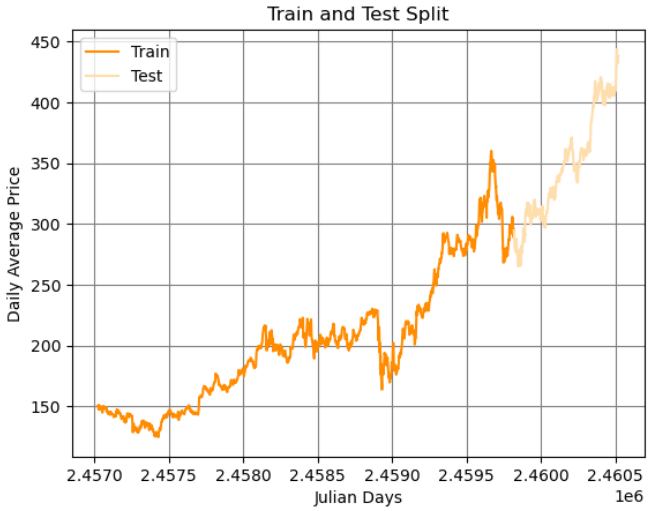

Fig. 2. The train (orange) and test (light orange) sections of the dataset.

and close price.

III. Baseline Solution

For the first half of this project, the baseline selected was a linear regression model, as the clear choice of a recurrent neural network is not a statistical learning model.

A. Selecting a Baseline

Although we use a linear regression model in this study, some alternative simplistic models can perform similarly. Although more complex than our model, polynomial regression is still much more straightforward than cutting-edge machine learning solutions. Like linear regression, these polynomials still grossly oversimplify the complex nature of stock data. Similarly, Lasso (or L1) regularization is also more advanced than linear regression but still underfits the data. Later, our baseline model will be compared to L1 regularization in their ability to predict future stock prices.

Linear regression was chosen over Lasso regularization and polynomial regression because of its simplicity. By definition, a baseline represents the simplest possible solution given the inherent constraints of the problem. We believe linear regression best meets this criterion, as both alternatives require additional computation.

B. Implementation

As illustrated by Fig. 3, our model has three main steps. First, we create an instance of Scikit-learn's LinearRegression class. This results in a variable representing an object with the weight and bias of our future line of best fit. Then, we fit this LinearRegression object to the training data. The final weight and bias of the model will be determined here. Last, we predict the future stock prices using the scaled Julian days within the testing data. An additional variable stores the predictions generated by the model.

We also predicted stock prices with Scikit-learn's Lasso regularization object. Implementing this model is nearly identical

```
# declare and train model
model = LinearRegression()
model.fit(X_train.reshape(-1,1), y_train)

# predict
y_pred = model.predict(X_test.reshape(-1,1))
```

Fig. 3. The implementation of the linear regression model.

to the linear regression model; however, this Lasso object requires additional parameters upon initialization. Namely, several different alpha (or regularization strength) values from 0.0001 to 1 were passed into the model via a one-dimensional grid optimization to shorten training time while maintaining performance. The performances of each can be seen in Fig. 4.

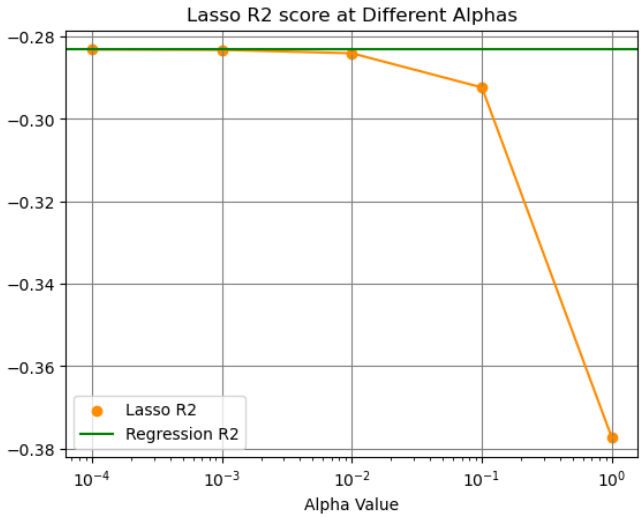

Fig. 4. The linear regression line of best fit.

C. Results

Since the model was a simple, statistical learning model, the linear regression and the lasso regularization results are not ideal. While following the assignment criteria, the baseline solution failed to effectively capture the underlying trends in the testing data, as shown by the negative R^2 score. This negative R^2 score indicates that the model is worse than just predicting the average of the target data. A large MSE score suggests that the model's predictions are less accurate.

As for the results of the grid optimization, it was observed that the linear regression model outperformed the lasso regularization each time. Fig. 5 portrays the lasso with a regularization strength of one (1). Decreasing the alpha would decrease the MSE and R^2 for the lasso regularization, but it would still not outperform the regression. However, at exceedingly low regularization strengths like one ten-thousandth (.0001), the difference in performance was negligible as seen in Fig. 6. The similar performance is unsurprising, as both the L1

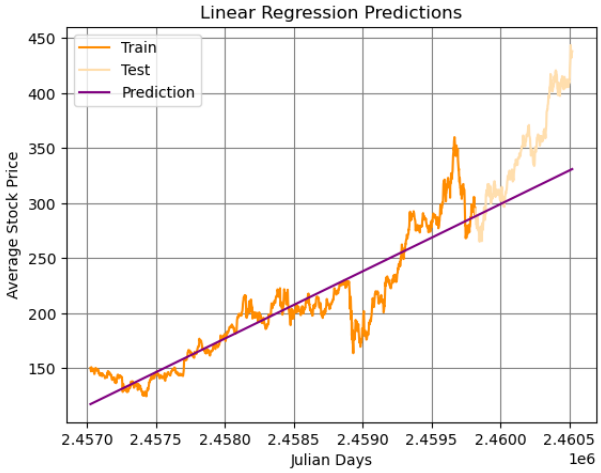

Fig. 5. The linear regression line of best fit.

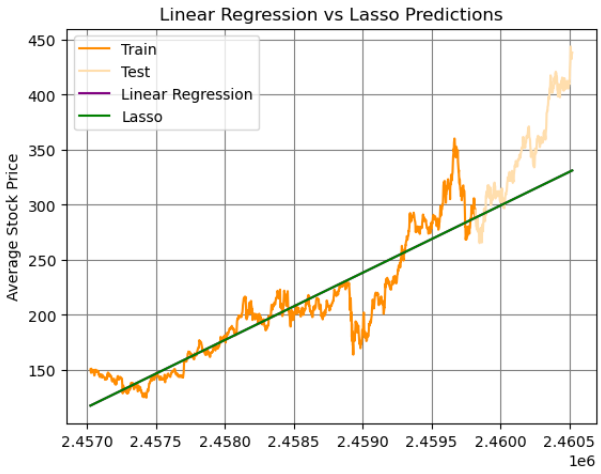

Fig. 6. The linear regression line of best fit and the lasso regularization line.

regularization and linear regression are linear models with a singular weight and bias.

These results suggest that lasso regularization may not have been suitable for this dataset, and a different regularization could have been used for an extension.

```
MSE:
        Regression: 2538.38
        Lasso: 2538.40

R2 Score:
        Regression: -0.2832
        Lasso: -0.2832
```

Fig. 7. The MSE and R^2 for Linear and Lasso Regularization respectively.

IV. Proposed Extension

Linear regression is not ideal for a stock prediction tool, but we can advance our product by adding new features and visualizations. It is important to note that stock predictions that are strictly based on past stock prices will not be effective, as there are many factors that affect stock prices. We propose a confidence interval for accuracy and a polynomial regression fit.

To quantify the model's reliability, a 95% confidence interval for accuracy can be used. This will offer a range in which the predicted prices will fall 95% of the time, which will allow assessment of uncertainty. A linear regression model will have a more significant level of uncertainty than a more ideal model.

A polynomial regression model could also be implemented to capture the non-linear nature of stock prices over time. The degree could be determined by testing multiple different degrees and selecting the degree with the lowest prediction error. We will evaluate the performance of the polynomial model the same way we assess the linear regression model, with root mean square error, mean squared error, and R-squared. In addition, we could apply the confidence interval to the polynomial and linear regression models to visualize the predictions better. Although the change in stock prices is not constant through time, that does not imply that a polynomial regression model will be better. Polynomial regression runs the risk of over-fitting and could be worse at predicting stock prices than the existing linear model.

V. Distribution of Work

Evan was responsible for the basic data curation. This involved loading the CSV, computing the price between open and close for a given day, and putting the date in terms of Julian days. Evan also led in organizing group meetings, creating tasks and issues, and writing the Baseline solution and Existing solution sections.

Beatrice split the dataset into training and testing subsets, applied a standard scaler to standardize the features for the future linear regression model analysis, and generated visualizations of the training and testing segments. She wrote the Results, Distribution of Work, and the reference sections.

Isaac developed and fit the Linear Regression and lasso models, and he found each one's performance. Isaac also made the markups in the Jupyter Notebook and wrote the Introduction paragraph.

VI. Finding Sources

Given that the scope of this assignment was confined to content learned from this course, there was no need to look up outside resources to help create the model. We used the theory we learned from class and applied it to the data as we did on our past homework assignments. During the research portion of our project, it became clear that a Long Short Term Memory recurrent neural network would be the best approach [2]. However, that was outside the scope of the class as it is a deep learning technique, so we adhered to the model we

chose along with the information and theory we had learned in class.

REFERENCES

[1] "What Percent Of Americans Own Stocks?," Financial Samurai, Jul. 09, 2019. https://www.financialsamurai.com/what-percent-of-americans-own-stocks/

[2] GunKurnia, "Stock Prediction Using the Best Algorithms: An Innovative Approach," Medium, Sep. 10, 2024. https://medium.com/@gunkurnia/stock-prediction-using-the-best-algorithms-an-innovative-approach-9740ca61daae (accessed Sept. 17, 2024).