HIGH SCHOOL EDITION
Journal of Student Research

Volume 14 Issue 1 (2025)

FAANG Stock Forecasting: Using Ridge Regression, Linear Regression, and Neural Networks

Zacharius Song

Northern Highlands Regional High School, USA

ABSTRACT

Blue-chip stocks serve as an important cornerstone in the stock market especially for long-term investors, since they specialize in high stability, high liquidity, and long-term growth. However, predicting future prices remains a challenge because of increased market volatility. This research aims to compare and evaluate the efficacy of three machine learning models: a ridge regression model, a linear regression model, and a neural network model— in order to forecast top-performing tech stock prices and trends over the long term. Using historical data (opening/closing prices, high/low prices) from Meta (formerly known as Facebook), Amazon, Apple, Netflix, and Alphabet (formerly known as Google), each of the three types of ML models will be developed for each stock. The models will all be separately trained and tested; they will be assessed for predictive accuracy using various success metrics (MSE, R^2, RMSE) and will be compared with each other using MAE as the common success metric. The neural networks had the MAE with the least value, with ridge regression having the greatest MAE value. Based on this MAE comparison, the research concluded that the LSTM recurrent neural networks had the most accurate outputs with minimal error, linear regression performed the second best and ridge regression performed the worst.

Methods

Linear Regression

Being one of the most common machine learning models, linear regression attempts to draw a linear relationship between two variables. However, it is very situational and one-dimensional: if there is no apparent linear relationship between variables, the model wouldn't be effective. On paper, linear regression would be effective for this research since the relationship between stock metrics would be linear over time. However because the data which is processed is real-world data and not a perfect simulation, it is possible for the model to lose accuracy. [2, 5, 7, 8]

Model Development

To train the linear regression model, the Linear Regression package itself has to be imported from the scikit-learn library. An independent variable and dependent variable also has to be created. We set the independent variable to be the opening price of the stock for the day and the dependent variable the closing price. Then, the dataset has to be divided into two segments: the model was set to have twenty percent of the data used for testing, while the remaining eighty percent for training. After divvying up the data, the variable name of the model itself was set to "regression_model". After successfully training the model, we assign all of the model-generated prediction outputs to the name "predictions". Finally, the R^2 value for the model was calculated to evaluate the model's efficacy, and MAE was calculated for comparison with the other models. This process was repeated for each stock.

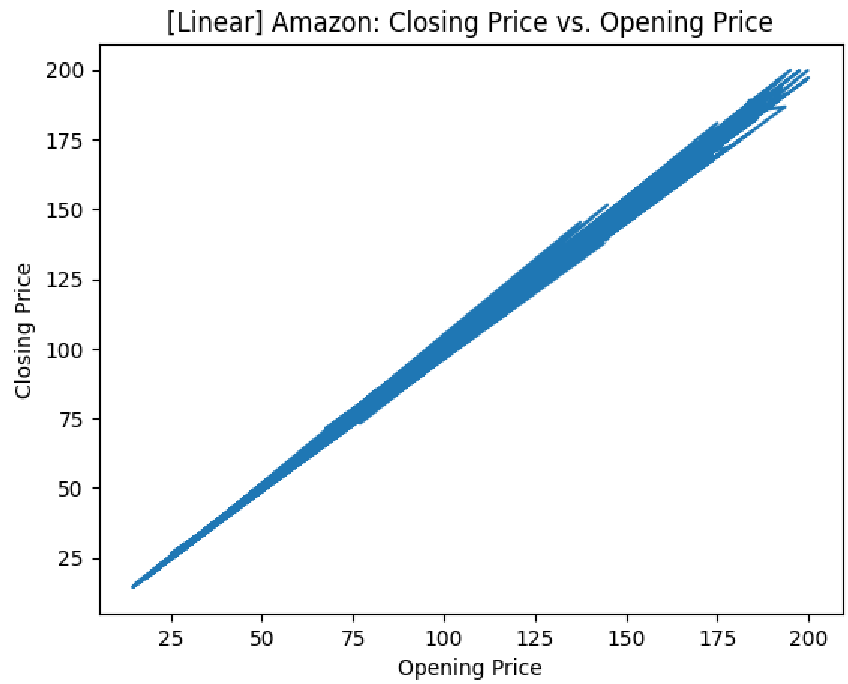

Figure 1. Linear Regression Graph for Amazon.

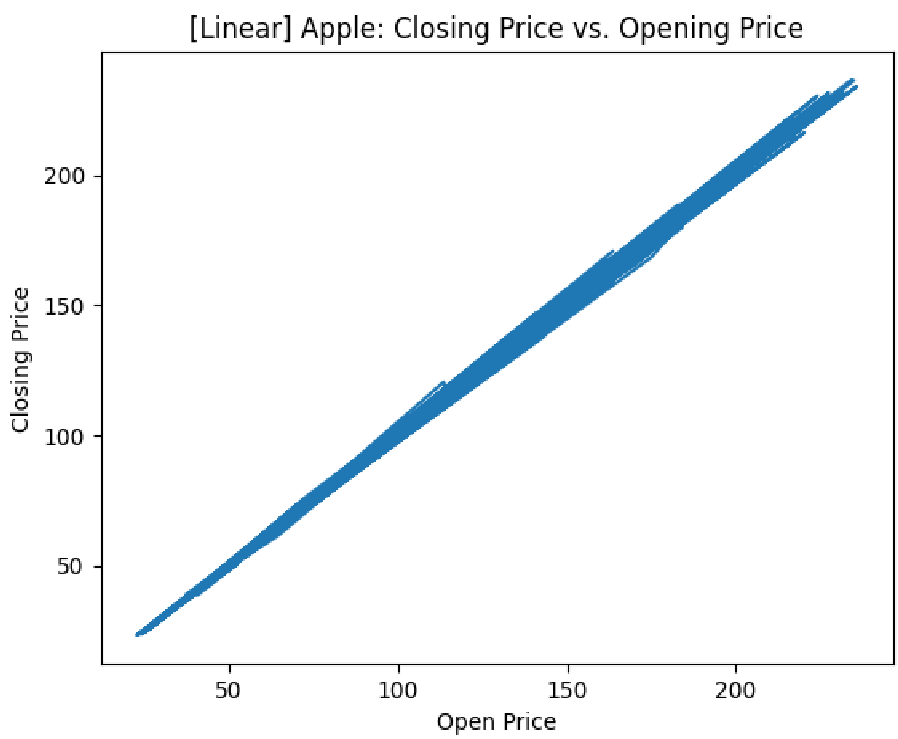

Figure 2. Linear Regression Graph for Apple.

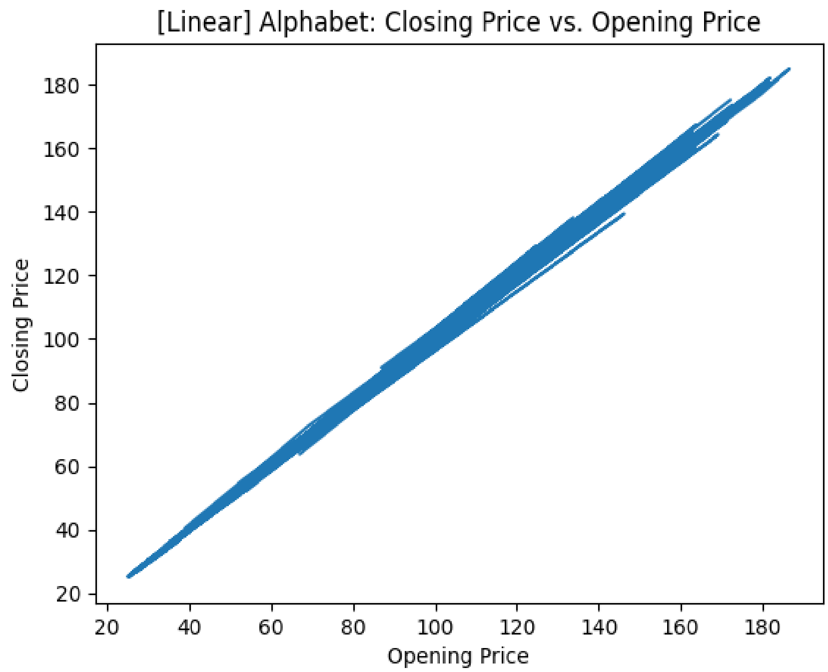

Figure 3. Linear Regression Graph for Alphabet (Google).

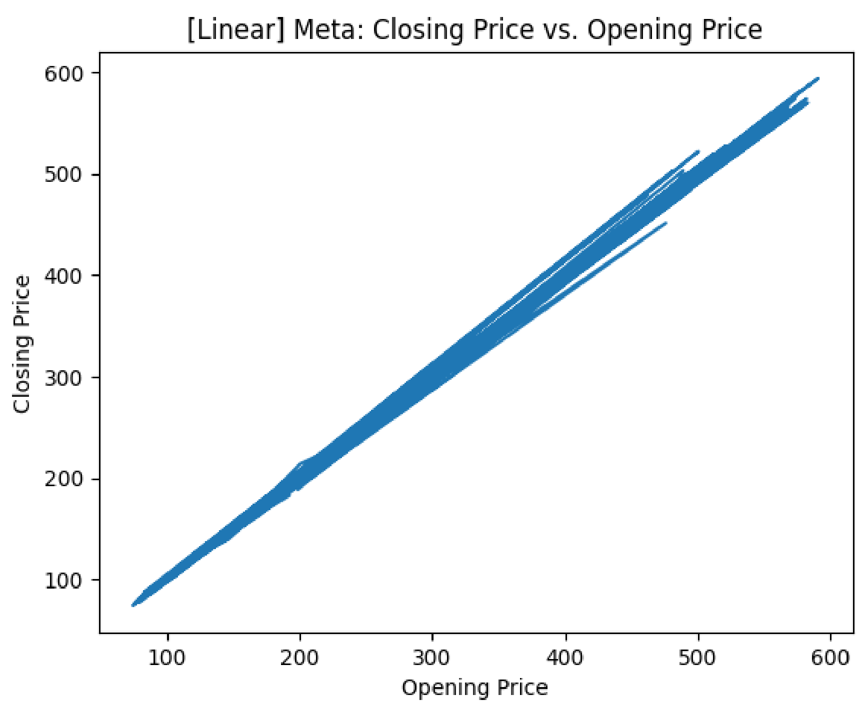

Figure 4. Linear Regression Graph for Meta (Facebook).

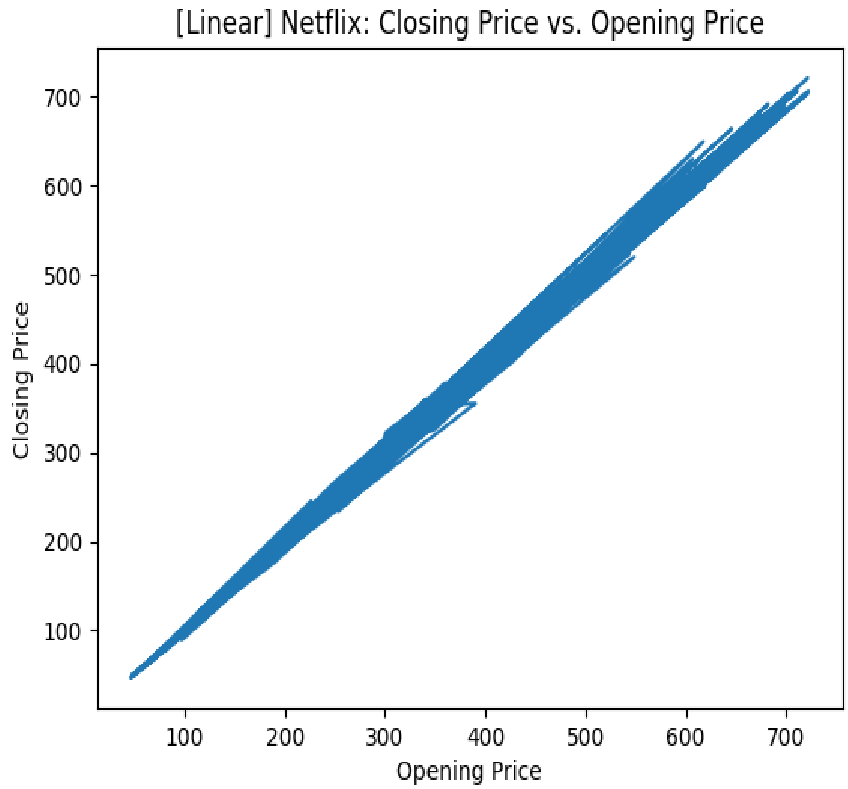

Figure 5. Linear Regression Graph for Netflix.

Ridge Regression

The ridge regression model is very similar to linear regression, in that they both produce straight, linear lines when modeled. However ridge regression has a mechanism to prevent overfitting. Overfitting is when the model takes in random fluctuations and the useless "outlier" points in the data, which leads to inaccuracies and fluctuations in outputs. [6, 7]

Model Development

The process to create a ridge regression model is very similar to the setup of a linear regression model. Like the linear regression process the dataset was divided up into twenty percent training and eighty percent testing, the MSE was calculated for the model's accuracy, and MAE was calculated for comparison with the other models. One of the big differences between a ridge regression model and a linear regression model is that hyperparameters are different for ridge regression. Most importantly, ridge regression utilizes a hyperparameter called alpha.

<u>Hyper Parameterization:</u> Hyper parameterization is the process of tuning and modifying parameters which aren't set by the training data in order to achieve maximal predictive accuracy. The alpha parameter is how much emphasis is placed on preventing overfitting. A higher alpha value implies the model is set to be more preventative of overfitting. A lower alpha value implies the model is set to be less preventative of overfitting. The most optimal alpha value for the ridge regression model for each stock will be shown. However, there are a limited number of alpha options, which are: 1e-15, 1e-10, 1e-8, 1e-3, 1e-2, 1,5, 10, 20, 30, 35, 40, 45, 50, 55, and 100.

HIGH SCHOOL EDITION
Journal of Student Research

Another part of the hyper parameterization process is recording the “best score”. The best score shows that the model with the best combination of hyperparameters produced the best accuracy score. However for the purpose of simplicity the research will only be focusing on R^2, MSE, and RMSE for results and comparison.

Figure 6. Ridge Regression Graph for Amazon.

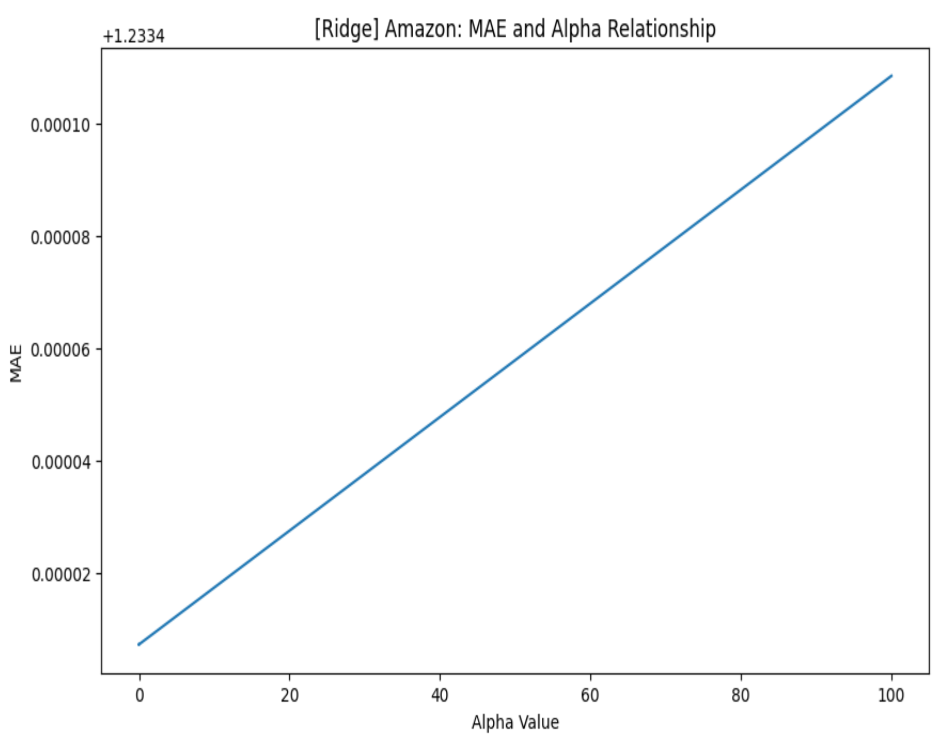

Figure 7. MAE and Alpha Relationship Graph for Amazon.

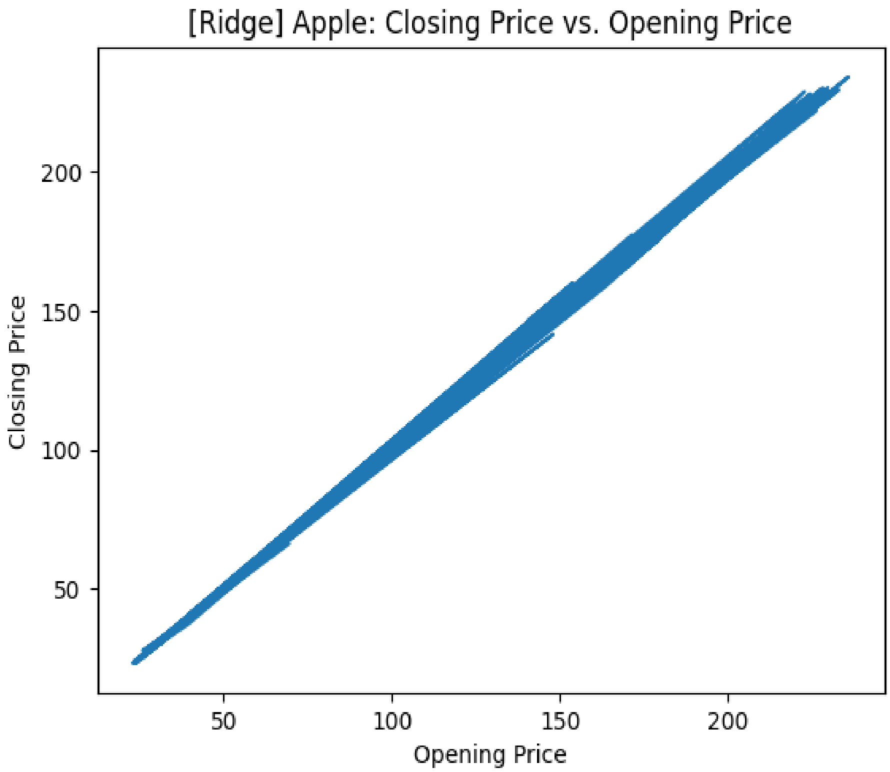

Figure 8. Ridge Regression Graph for Apple.

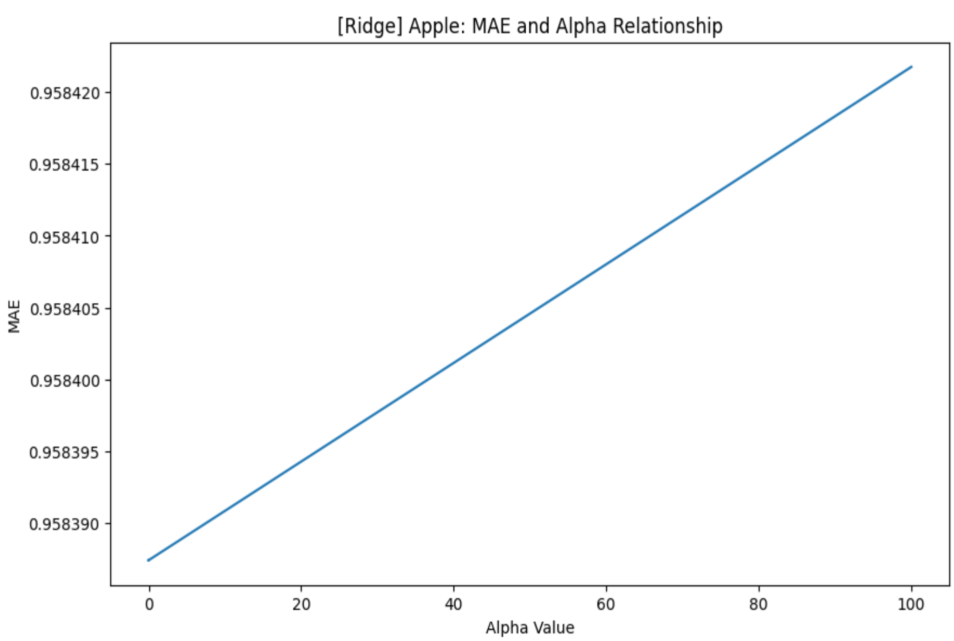

Figure 9. MAE and Alpha Relationship Graph for Apple.

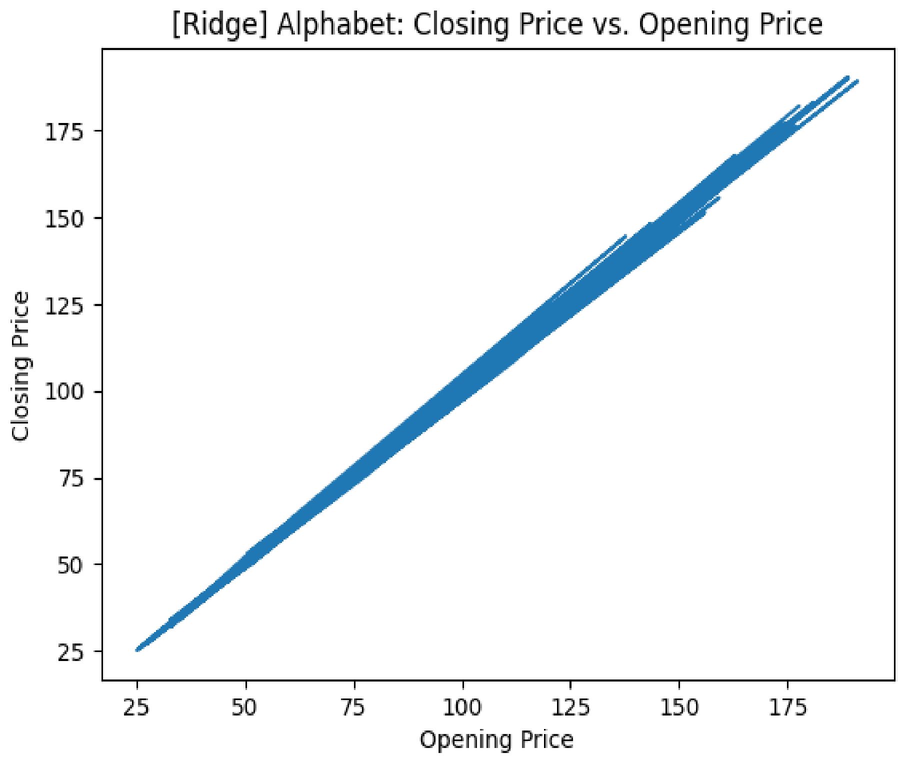

Figure 10. Ridge Regression Graph for Alphabet (Google).

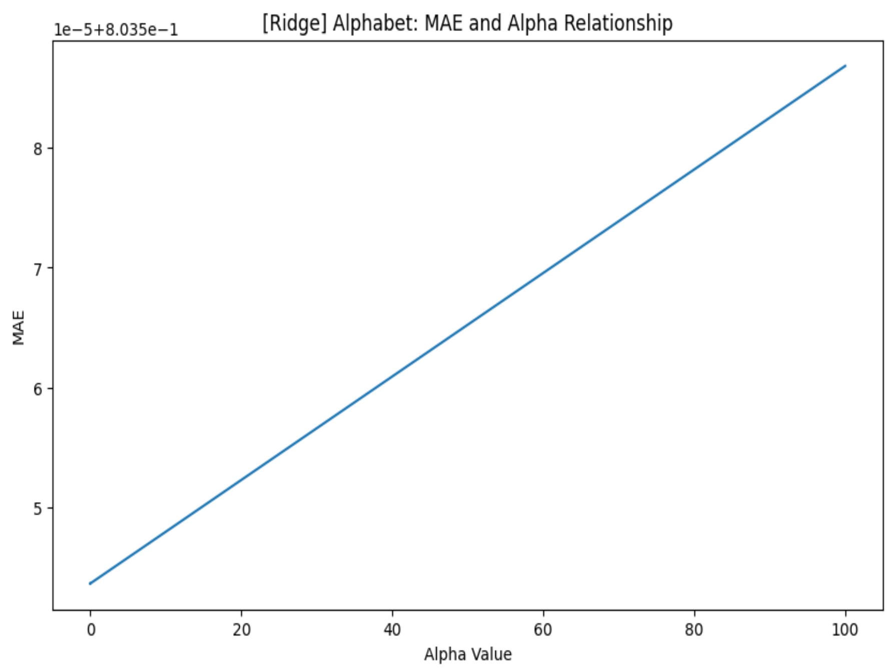

Figure 11. MAE and Alpha Relationship Graph for Alphabet (Google).

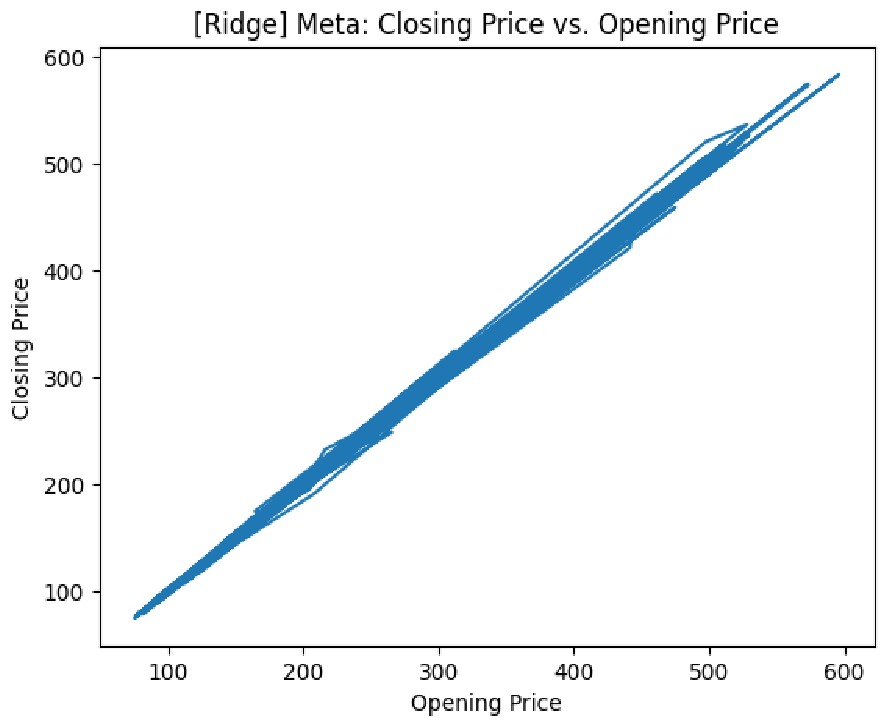

Figure 12. Ridge Regression Graph for Meta (Facebook).

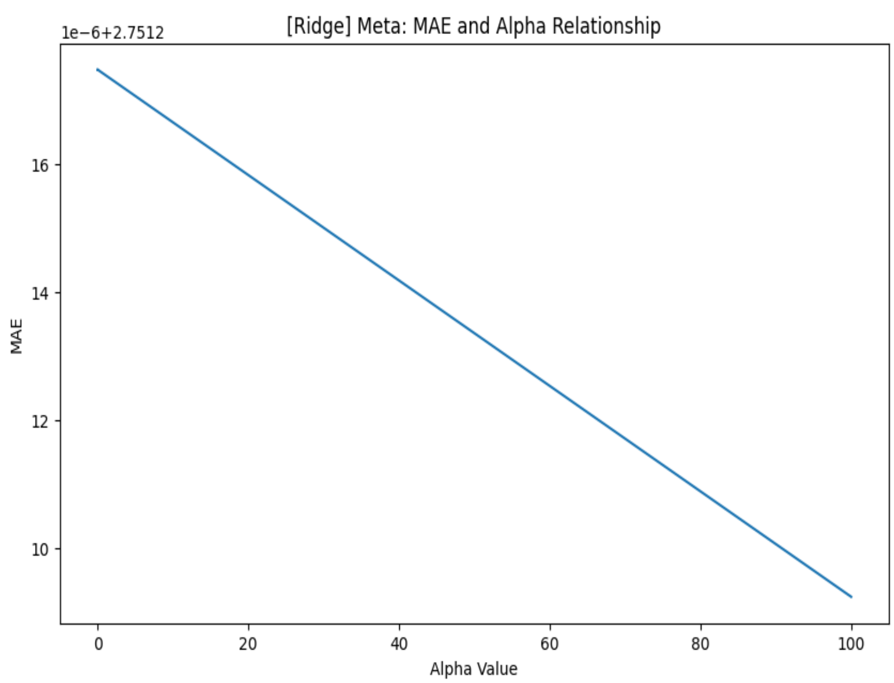

Figure 13. MAE and Alpha Relationship Graph for Meta (Facebook).

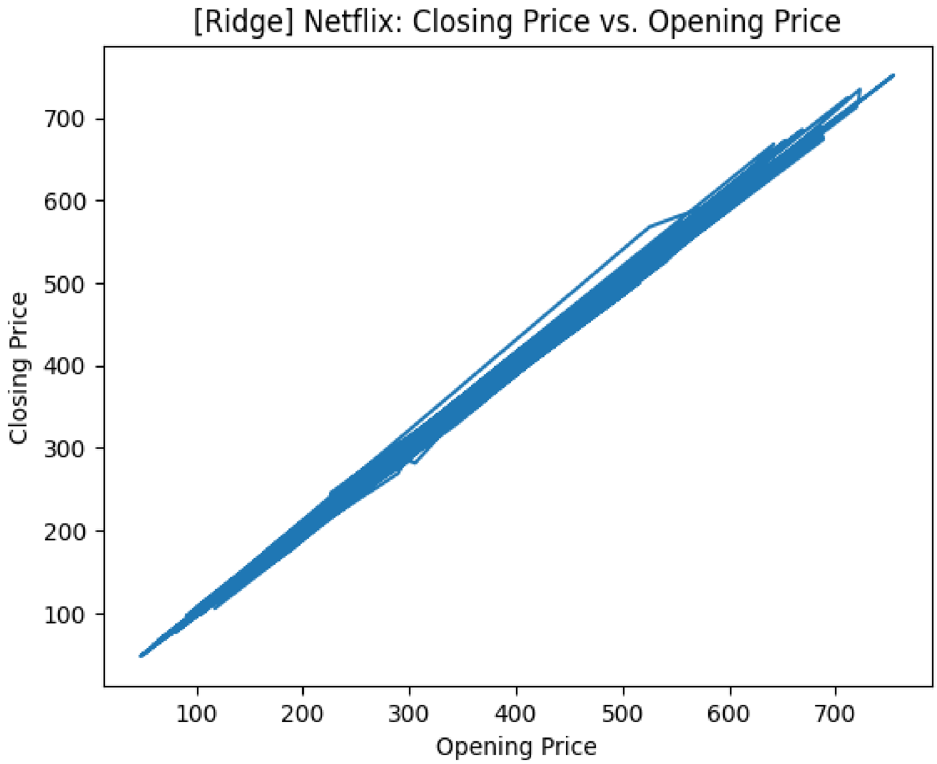

Figure 14. Ridge Regression Graph for Netflix.

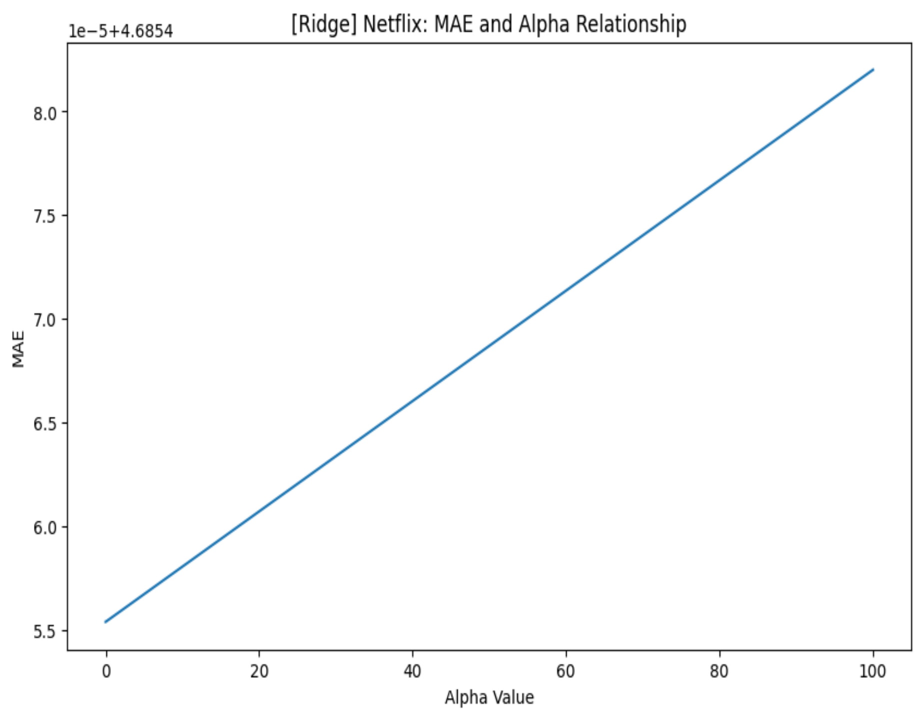

Figure 15. MAE and Alpha Relationship Graph for Netflix.

Neural Networks

Neural networks is another common machine learning model type, yet it is very different compared to linear regression and ridge regression models. Neural Networks can handle all different types of inputs, whether it be NLP (natural language processing), numerical data, photos, text, and more.

The neuron is the fundamental building block of a neural network model. They each hold on to one specific feature, or important piece of data, at a time. The neurons are organized into different layers, where the neurons behave differently for each layer. The input layer is where the data first enters. The output layer is the last layer of the neural network, where outputs are sent out. In between these two layers are the hidden layers. These are the layers that do all of the processing.

Usually, data passes through the layers in a forward motion, AKA feedforward neural networks. The problem with this is that these conventional neural networks have a complication called the Vanishing Gradient Problem, where models could not retain and process data over a larger time-window. Essentially, they couldn't build off of previous data and establish new patterns and connections. This is where the Recurrent Neural Network comes in: the type of Neural Network used for this research.

RNNs are capable of learning and remembering what previous inputs– this is called forecasting sliding window. Forecasting sliding window allows for the examination of the effect of multiple dates' worth of data on the future stock prices, so that the model's prediction is based on seasonal/time-based patterns in addition to previous values. [1, 3, 4]

LSTMs

LSTMs are a type of recurrent neural network. They specialize in remembering data for long periods since other neural networks struggle with retaining, adding, or removing data over time. This makes LSTMs especially effective for projects in bioinformatics, natural language processing, and most importantly finance.

Model Development and Hyper Parameterization: To start off, we set the training data size to ninety-five percent of the data. This is because for neural networks, you want a bigger training set since recurrent neural networks thrive when given lengthy inputs. When you construct the model, there are hyperparameters you can modify: number of filters, kernel size, stride, ReLU, and density. Filters and kernels are two words for the same model component, they are small matrices to detect the most important patterns from the input data. The size of a kernel is essentially the sample size. More filters means that they can capture wider varieties of data from the input data. Finally, the "dense" hyperparameter connects neurons together from layer to layer to allow for better analysis of complex patterns from the input data. Following this, a standard Adam optimizer was used as well as a Hubert loss function. The Adam optimizer allows the model to use adaptive learning rates and bias correction for more efficiency and accuracy. The Hubert loss function is another hyperparameter that specializes in eliminating data outliers to skew outputs. Finally, the accuracy of these LSTM models were evaluated by using RMSE, a success metric for specifically neural networks, similar to how MSE was used for ridge regression and R^2 for linear regression. And just like the other two neural network types, MAE was calculated for comparison with the other models.

Figure 16. Neural Network LSTM Graphs For Each Stock.

Results

Linear Regression Success Metrics	R^2 Value	MAE Value
Amazon	0.9987821220747459	1.2205
Apple	0.999387782374249	0.9259
Alphabet (Google)	0.9992385225018652	0.8103
Meta (Facebook)	0.9987638311205832	2.6744

Netflix	0.9984583835633928	4.6315
Average	0.9989261283	5.1313

Figure 17. Linear Regression Success Metric Table.

Ridge Regression Success Metrics	MSE Value	MAE Value
Amazon	5028.103024752435	1.1869
Apple	8067.950344601414	0.9667
Alphabet (Google)	1.4553688101585434	0.8128
Meta (Facebook)	24796.557000220146	2.6083
Netflix	69009.93615430135	4.7692
Average	21380.80038	5.17195

Figure 18. Ridge Regression Success Metric Table.

Neural Network Success Metrics	RMSE Value (dollars)	MAE Value
Amazon	16.43184155115719	0.0137
Apple	5.085945958074165	0.0152
Alphabet (Google)	9.70701014241026	0.0201
Meta (Facebook)	5.057182193065304	0.0190
Netflix	4.49502942797078	0.0130
Average	8.155401855	0.0162

Figure 19. Neural Network Success Metric Table.

Discussion

Linear Regression

The linear regression models all performed very well, as each of the models produced an R^2 value very close to one. The closer an R^2 value is to one means that the model produced an almost perfectly accurate fit, which is what every linear regression model did no matter the stock.

However, the MAE values were inconsistent from stock to stock. Netflix and Meta had very high MAE values compared to Amazon and Apple, whose values were a lot closer to the neural network MAE values.

Ridge Regression

The MSE values were extremely sporadic as well, they were not consistent at all. To evaluate a model's predictive accuracy using MSE, one needs to know that a higher MSE equates to larger errors in prediction. Ideally, the model would have an MSE value of zero, since that would mean the model is perfectly accurate. For some of the stocks, the MSE was not close to perfection as some of the highest MSE values were in the tens of thousands. However one thing to note is that MSE values are normally higher than MAE values because MSE penalizes errors due to squaring.

The MAE values of the ridge regression models were only a little higher than those of the linear regression models. Like the MSE results, the MAE values were not consistent; similar to the linear regression MAE values, the ridge regression ones increased drastically down the table, leading to a high average.

LSTMs (Neural Networks)

The RMSE values of the neural networks were relatively very small, which signifies high accuracy with little error. For comparison, RMSE, although not identical to MSE, also amplifies errors due to squaring; however the LSTM models produced results far less in value than ridge regression MSE values.

As for the MAE values, they are extremely low, with an average far less than the MAE averages of both linear regression and ridge regression. They are also fairly consistent as well with no big fluctuations from stock to stock.

Conclusion

Using the MAE success metric to compare the three models, the RNN had the lowest MAE, followed by linear regression, with ridge regression having the highest value. The LSTM recurrent neural network models produced the most accurate results compared to the ridge regression and linear regression models for all five stocks given. This makes sense as neural networks perform the best with data retention in order to make better connections and more accurate predictions. This is because of the forecasting sliding window, which mitigates the Vanishing Gradient Problem other models have. When predicting future stock prices based on purely technical data, recurrent neural networks are the best model to use compared to linear regression and ridge regression models.

Limitations

Although nobody can look into the future, the best way of predicting it is by looking back into the past. That is exactly what this ridge regression model does— from the history of FAANG opening price, closing price, adjusted closing price, high price, low price, and volume, the model should make accurate stock forecasts in the future. However, nobody can look into the future. Although impactful geopolitical and economic events do have an influence on the metrics provided to the model, the model cannot fully anticipate nor understand them since it specializes only in technical analysis. Therefore if any big event occurs in the near future— which, looking at current national and global tensions right now, is almost undeniable— the model's projections will be rendered inaccurate and unusable.

Acknowledgments

I would like to thank my advisor for the valuable insight provided to me on this topic.

References

[1] Colah, J. (2015, August 27). *Understanding LSTM networks*. http://colah.github.io/posts/2015-08-Understanding-LSTMs/

[2] Yale Department of Statistics. (1997). Lecture notes: Simple linear regression. http://www.stat.yale.edu/Courses/1997-98/101/linreg.htm

[3] Patel, J., Shah, S., Thakkar, P., & Kotecha, K. (2015). Predicting stock and stock price index movement using Trend Deterministic Data Preparation and machine learning techniques. Expert Systems with Applications, 42(1), 259–268. doi:10.1016/j.eswa.2014.07.040

[4] Haykin, S. (1999). Neural networks: A comprehensive foundation (2nd ed.). Prentice Hall.

[5] Kuchibhotla, A. K., Patil, P., & Rinaldo, A. (2019). All of linear regression. arXiv preprint arXiv:1910.06386. https://doi.org/10.48550/arXiv.1910.06386

[6] Arashi, M., Roozbeh, M., Hamzah, N. A., & Gasparini, M. (2021). Ridge regression and its applications in genetic studies. PLOS ONE, 16(2), e0245376. https://doi.org/10.1371/journal.pone.0245376

[7] S. Ansari and A. B. Nassif, "A Comprehensive Study of Regression Analysis and the Existing Techniques," 2022 Advances in Science and Engineering Technology International Conferences (ASET), Dubai, United Arab Emirates, 2022, pp. 1-10, doi: 10.1109/ASET53988.2022.9734973.

[8] M. Huang, "Theory and Implementation of linear regression," 2020 International Conference on Computer Vision, Image and Deep Learning (CVIDL), Chongqing, China, 2020, pp. 210-217, doi: 10.1109/CVIDL51233.2020.00-99.