Predicting Stock Closing Prices of MAAGA Companies using LSTM and Linear Regression

Yasmin Moule
College of Letters and Sciences: Cognitive Science and Applied Math
University of California Davis
Davis, CA, USA
ysmoule@ucdavis.edu

Yihan Niu
College of Letters and Sciences: Economics and Math Analytics and Operation Research
University of California Davis
Davis, CA, USA
yhniu@ucdavis.edu

***Abstract*—The stock market is commonly used in economics and related fields to predict financial markets. Since President Trump took office for the second time in January 2025, he has imposed tariffs on many different industries. Among other things, he imposed tariffs on technology. To see the effect of these tariffs, we investigated trends in the technology market through stock market analysis and created two machine learning models to predict future stock market prices. We investigated the technology market by analyzing the largest technology companies (Amazon, Apple, Google, Meta, and Netflix), collectively known as MAANG.**

We created a linear regression model and a long-short-term memory (LSTM) model. This paper highlighted the methodology used and compared the accuracy of the two models. Finally, this paper used the models that predict the next two months of stock prices.

***Index Terms*—Machine learning, stock price prediction, linear regression, long- and short-term memory model, MAAGA Companies**

I. Introduction

The stock market is a complex and dynamic marketplace where publicly traded companies' stocks are bought and sold. It serves as a platform for companies to raise capital by issuing shares and for investors to purchase those shares in the hope of earning returns. The stock market is influenced by a multitude of factors, including economic indicators (such as GDP growth, inflation rates, corporate earnings reports, geopolitical events, and investor sentiment) [1]. Predicting the price of the stock can let investors make informed decisions about when to buy, hold, or sell their stocks.

Since April 12, 2025, there has been a new tariff war between China and the United States. This project was centered on an in - depth analysis of the stock market performance of several large American companies. We collected stock market data of the prominent MAANG group (Meta, Amazon, Apple, Netflix, Google - Alphabet) starting from their founding up to the present. By scrutinizing period from March 2025 to now, we gleaned some insight into how the tariffs are influencing the stock prices. Subsequently, we leveraged the analyzed data to predict the stock price trends of the chosen companies over the next two - month span from May to July. This forecast was grounded in a blend of historical data analysis, machine learning, and linear regression based on data availability and relevance, aiming to offer a well - founded projection to aid investors, financial analysts, and other stakeholders in making more strategic stock - market decisions.

Among other things, this paper includes:

- An exploratory data analysis (EDA) process to identify relationships between companies and stock price trends.
- Development of a linear regression model and a LSTM model to compare performance.
- Evaluation of the aforementioned models to determine the most accurate model for stock prices.
- Prediction of future stock prices and analysis of results

II. Literature Review

A. Stock market's price movement prediction with LSTM neural networks

This paper, authored by Nelson, Pereira, and de Oliveira, discusses use of LSTM in stock market prediction [5]. It explains that LSTMs are traditionally used in Natural Language Processing (NLP) because they sequentially process each word, one at a time. Thus, a task might be predicting stock prices based on newspaper clippings [5]. It also introduces various economic hypotheses about the stock market. First, the Efficient Market Hypothesis claims that a stock price reflects all previous information available for it. Thus, there's no available information can be used to predict stock prices, only new, unexpected information changes stock prices. On the other end of the spectrum, the Random-walk hypothesis claims that a stock price changes independently of its history. Both hypothesis claim that stock prices cannot be accurately predicted [5] . The authors of this paper attempt to address this claim.

The authors also created a LSTM model with high performance. Their model was retrained each day and predicted 15 minutes into the future [5]. They predicted if the stock price would go up (1) or not (0) and tested their model on real stock data, where if the model outputted 1, they would buy a stock and sell it 15 minutes later [5]. This resulted in a net profit. Their paper shows that it is possible to have LSTMs with high performance on the stock prediction task.

B. Research on Multistep Time Series Prediction Based on LSTM

This paper, by Wang, Zhu, and Li, describes two different methods using for multistep time series prediction for LSTM. The first method is Multi-step input, in which the input is a vector of $(x_{t-(m-1)}, \ldots, x_t)$ to predict $(x_{t+1}, \ldots, x_{t+n})$ [9]. They then put the data through two LSTM layers with 100 cells each and one dense layer before generating a vector of predictions. By varying the length of the input vector, they found that when the input time step is closer to the seasonal period of the dataset, the better the model performs [9]. We also attempted to use this fact for our model, setting the input time step to 7 to represent a week. Unfortunately, we later found that the stock market week does not include weekends and is thus 5 days long. Our models still performed better with 7 as input time step, so we kept this.

The other method paper covered is called seq2vec. In this method, the input vector is inputed to the model one by one and the LSTM layer includes only one cell [9]. For each input, the LSTM layer outputs a prediction, then finally a Dense layer combines the predictions to create the final prediction vector. This method had similar error at the first predicted time step, but performed much better at the later time steps than the other method.

Unlike our models, these methods define models that create a vector of predictions. In contrast, our models generate one prediction, add it to the input vector, and predict another closing price. Because at the core our models are created for single step prediction, we may find greater error when applying our models to multi-step prediction than if we had created a model like the ones in this paper.

III. Dataset Description

The data was taken from Kaggle (https://www.kaggle.com/datasets/nikhil1e9/netflix-stock-price). This dataset records stock prices from MAANG companies (Meta, Amazon, Apple, Netflix, and Google) from the day they were founded until May 6th, 2025, which is when we downloaded the dataset. The dataset includes the values of the date the stock price was recorded, the opening and closing prices of the stock for that day, the high and low prices for that day, and the total number of shares traded on that day. It includes separate csv files for each company. It also includes separate csv files for data recorded daily, weekly, and monthly to allow for analysis of the stock market both in the long-term and short term. We used only the daily files in this project.

IV. Exploratory Data Analysis

Before beginning with our models, we first analyzed the dataset to see trends in the stock market and among the individual companies using historical data.

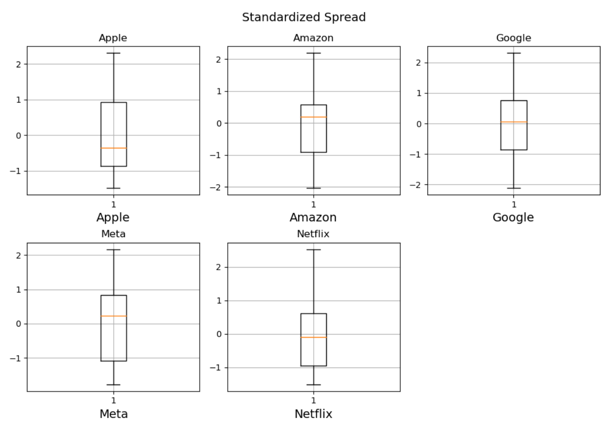

Fig. 1. Box plot of Closing Prices over last 500 days

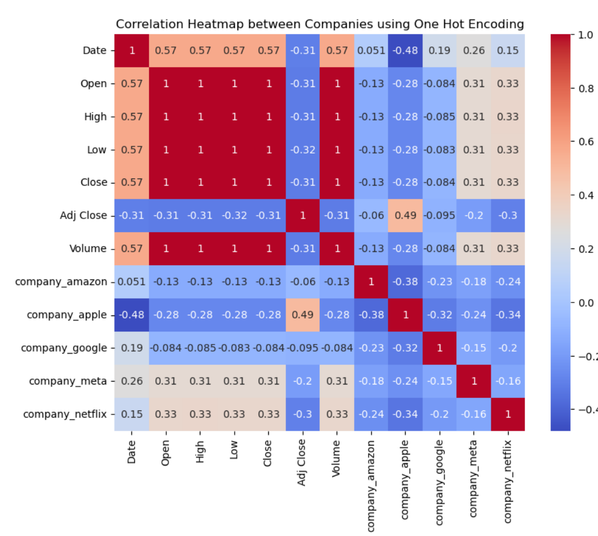

Fig. 2. Correlation Matrix made using Weekly Data

Figure 1 shows the standardized (using Z-score) box plots of the closing prices for each company over the last 500 days. There are no outliers in this data. (Note that if you create box plots for all the data, not just the last 500 days,

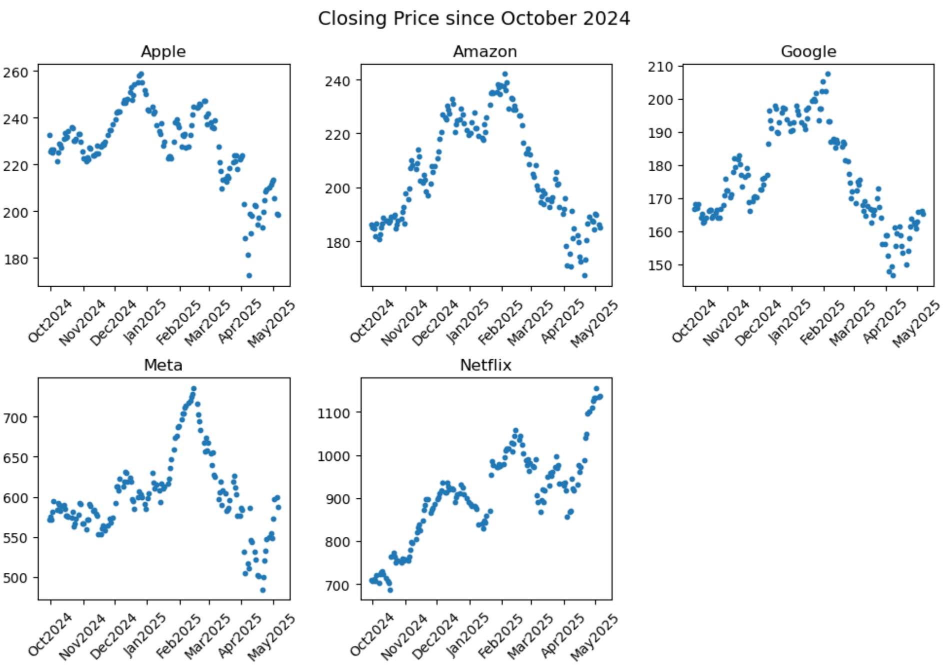

Fig. 3. Closing Price since Trump's Inauguration using data gathered daily

you have many upper outliers, likely due to inflation). Apple is slightly skewed to the left while Amazon and Meta are lightly skewed to the right. Google and Netflix are mostly symmetrical. All box plots have similar spread, indicating that the closing stock prices for the different companies vary by similar amounts.

By creating a Correlation matrix between the features of the data (Figure 2), we see that the features high, low, open, close, and volume are perfectly correlated. Thus, we can use any of these features to predict stock market trends. The moderate positive association between stock prices and date shows that stock prices overall have gone up over time. This may be due to inflation since the correlation matrix contains data from the founding of the companies until now.

Figure 3 shows the closing price of stocks over the past two years. For most companies, we see a decrease in closing price since January (i.e. since Trump was inaugurated). Netflix, although they experienced a short decrease, has a mostly upwards trend. We expect our models to capture these trends as well as predict the future value of these stocks, given that they are currently changing a lot due to the political environment.

V. Proposed Methodology

A. Overview of Proposed Methodology

In this section, we present the methodology used for predicting the closing stock prices. We utilized two different models, a Long Short Term Memory (LSTM) model and a linear regression model. They were chosen because they are very different models. The linear regression model is very simple and the LSTM model is very complex. Depending on which model is more accurate, we can make claims about the complexity of the relationship between previous stock prices and future stock prices. A more detailed methodology for each model is available below.

B. Linear Regression Model

Linear Regression process all inputs at the same time. Each input is multiplied by a weight and then summed to produce the output. Linear Regression assumes the data is linearly related and finds the line of best fit for the data. An advantage of linear regression is that unlike neural networks, the weights for each input often have interpretable meaning.

However, this approach has several drawbacks. Linear regression assumes a linear relationship between variables, which often does not hold true in the complex stock market where relationships can be highly nonlinear. It is also sensitive to outliers, which are common in stock price data due to unexpected events. Additionally, the model struggles to capture complex patterns like seasonality and regime switching, and there is a risk of overfitting if too many variables are included. Finally, it lacks the ability to adapt well to changing market conditions, as relationships between factors that influence stock prices can evolve over time.

C. Long Short Term Memory (LSTM) Model

Unlike the traditional (vanilla) RNNs, LSTMs avoid the exploding/vanishing gradient problem by storing an updating a long term and a short term memory [8]. LSTMs work by

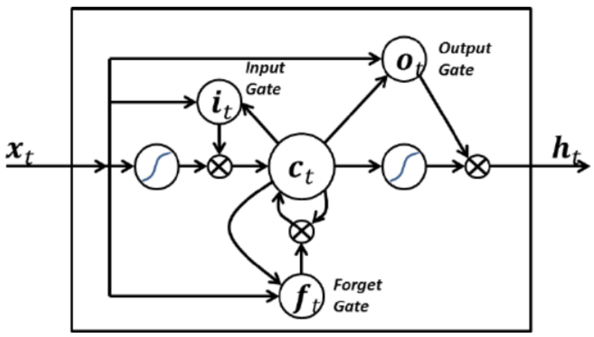

Fig. 4. LSTM Cell

processing input data sequentially, storing a portion of the input data as "long-term" memory and a portion as short term. Much like their name, the long-term memory keeps track of inputs from the beginning of the sequence the LSTM is processing while the short-term memory keeps track of the previous element. They then combine to form a prediction. This model architecture (Figure 4) works well with stock data, which is time-sequence data.

There are also few drawbacks of using the LSTM to train our model. First, LSTMs are computationally costly [7]. Because of this, it could be difficult to work with the entire large dataset at once. To address this, we will train our model on a more recent subset of the data rather than all of the data. This will address another common LSTM problem, which is that they have trouble with very long-range dependencies [7]. Since we will work with more recent data, we don't expect this to be a problem.

Here are a few more challenges that may arise with the LSTM model. The model is prone to overfitting the data, occasionally instable in training, and sensitive to hyper parameters [7]. To address the first issue, we will use dropout regularization. The second issue occurs with inadequate initialization, high learning rates, or noisy data. We will regularize the data via the MinMax scaler to help mitigate the effects of noise. We will also experiment to find adequate learning rates. Finally, we will use grid search to select suitable hyper-parameters for our model. Grid search will tune number of layers and regularization parameters, among others. The Adaptive Moment Estimation (Adam) optimizer will be used as our optimization function, which adjusts learning rates and momentum based on past gradients to avoid getting stuck at local minima and ensuring a smooth path to the optimal solution.

VI. Experimental Results and Evaluations"

A. *Linear Regression Model*

We implemented the linear regression model in Google Colab with Python libraries NumPy, pandas, and scikit-learn. The objective was to implement a stock price prediction system based on the stock closing prices of major companies over the previous 7 days to predict the stock prices of each day in the next two months.

We started by importing historical stock data for Amazon, Apple, Google, Meta, and Netflix, with the most recent date being May 6, 2025, the day we obtained the data. We then keep only the date and closing price columns and convert them to datetime type, filtered out data before year 2013 to focus on recent market dynamics. For each company's dataset, we engineered lagged features representing the previous seven days' closing prices (i.e., $t-1$ to $t-7$) to capture temporal dependencies. An important note is that stock market data is collected daily, excluding weekends, so a week is about 5 days. Thus, we trained on about a week and a half of previous price. We removed rows with missing values introduced by lagging, and applied one-hot encoding to the company names to create binary indicator variables. We then processed the data sets to allow them to concatenate into a single data frame, and Min-Max scaling (range [0,1]) was applied to both the feature matrix (the lagged prices) and the target variable (closing price) to normalize scales and enhance model convergence.

Data were partitioned using an 80:20 split, with 80% of each company's data allocated to training and 20% to testing, preserving temporal order to avoid data leakage. A linear regression model was trained on the combined training data, and the performance was evaluated using mean squared error (MSE), root mean squared error (RMSE), mean absolute error (MAE) and R^2.

Finally, actual training set prices and test set predictions were visualized via line plots, with evaluation metrics overlaid for clarity, facilitating interpretation of the model's predictive accuracy and trend capture capability (Figures 5-9).

Fig. 5. Amazon Stock Price Prediction

Based on the graphs, we can see that that this model generates MSE, RMSE, MAE and R^2 accordingly.

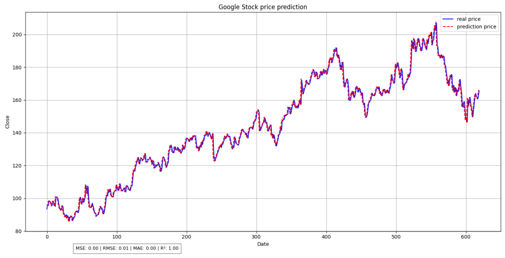

Fig. 6. Google Stock Price Prediction

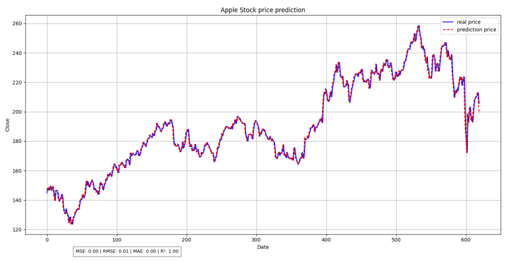

Fig. 7. Apple Stock Price Prediction

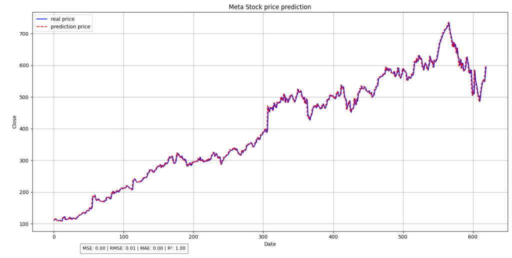

Fig. 8. Meta Stock Price Prediction

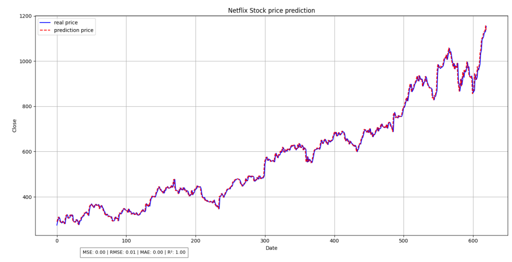

Fig. 9. Netflix Stock Price Prediction

Specifically, MSE is nearly 0.0000483416, RMSE is 0.006952809603903938, MAE is 0.0038590882602448765 and R^2 is 0.998584869145701.

These results show that model performs very well. First, MSE, RSE, and MAE are all very low values, indicating that the model prediction results are very close to the true values with small errors. Second,R^2 is 0.998584869145701 which indicates that the model almost completely captures the changing trend of the data and the fitting effect is very good. In addition, MAE is less than RSE, indicating that the model is less sensitive to outliers, the error distribution is uniform, and it shows strong robustness.

B. *Long Short Term Memory (LSTM) Model*

The LSTM Model was created using PyTorch. The basic model was created following a tutorial on Youtube [3]. The model was then altered to include multiple companies.

The data was preprocessed the same for the LSTM as it was for Linear Regression (i.e. Closing price as the target variable, the previous 7 days and one-hot encoded company labels as the input to the model). The closing stock prices were normalized to be between 0 and 1. Only the stock prices since 2013 were used, and the training/testing datasets were created using a 80/20 split. An important note is that the testing dataset contained the most recent 20% of data (i.e. $\sim 2023-2025$), meaning that the model has not trained on the most recent dates. This may affect its accuracy in predicting future stock prices.

Hyperparameter tuning was performed via Grid Search. The general model structure can be seen in Figure 10.

Hyperparameter tuning was performed on many paramaters, which can be seen in table I. The model with the lowest mean squared error had parameters: Batch size = 32, Learning Rate = 0.001, Dropout = 0.2, LSTM Hidden Size = 32, 1 linear layers, 1 stacked layer, and a regularization parameter of 0.001 for Ridge (L2) Regularization. It's important to note that since there was only one LSTM layer, dropout was automatically set to 0 to avoid dropping the input neurons. The optimal number of epochs of training was determined by graphing epochs vs MSE and was chosen to be 3 epochs.

Learning rate	Batch size	LSTM layer Hidden Size	Number of LSTM Layers
0.001	16	4	1
0.01	32	32	2

Number of Linear Layers	Regularization parameter	Dropout
1	0 .001	0.2
2	0	0.5

TABLE I
HYPER-PARAMETER GRID

The final model was retrained with the above parameters for 3 epochs. It performed well overall, with MSE: 0.0007223, R^2: 0.9740, RMSE: 0.02688 and MAE: 0.01450. Then output of the model was then inverse transformed to be back within its original scale. The model generated fairly accurate predictions on the testing data, shown in Figure 11.

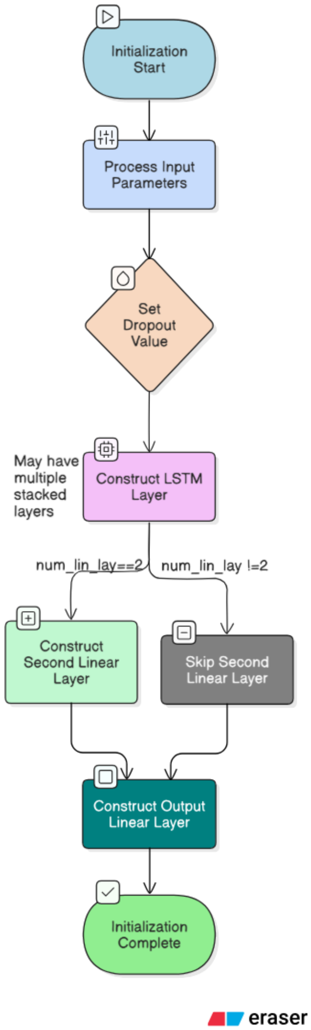

Fig. 10. LSTM Model Structure

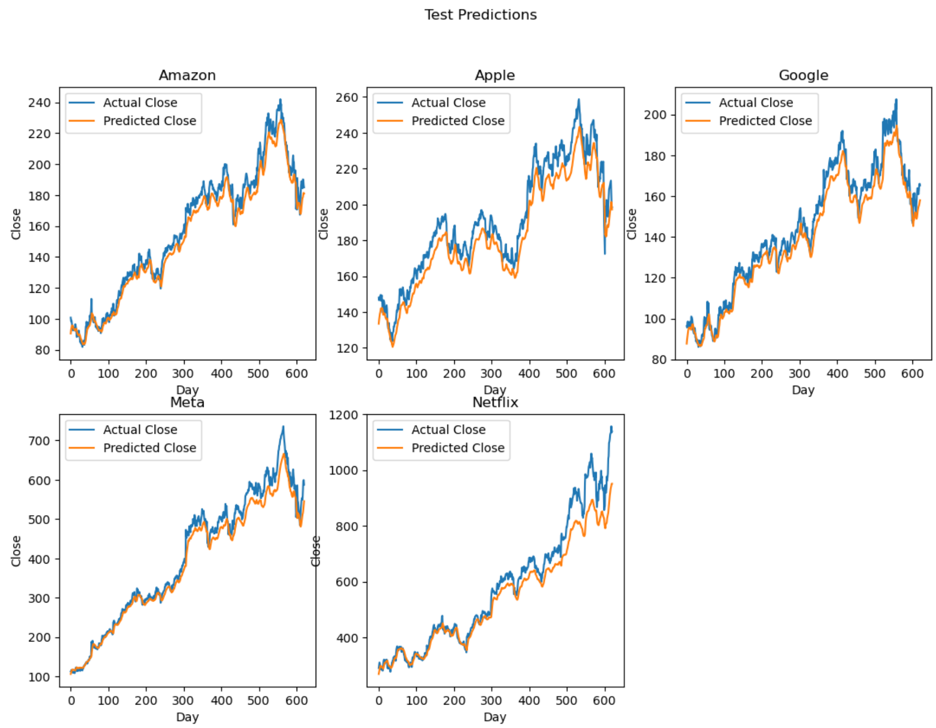

Fig. 11. LSTM Model Predictions on Testing Data

C. Comparing the two models

Although the two models both performed well on the testing set, it is clear by comparing the accuracy scores that linear regression is the better model (Table II).

Model	MSE	RMSE	MAE	R^2
Linear Regression	0.00004832	0.006953	0.003860	0.9986
LSTM	0.0007223	0.02688	0.01450	0.9740

TABLE II
COMPARING MODEL SCORES

We can also see this visually. When comparing the prediction vs true price graphs for each of the companies, it is clear that linear regression matches the true price better, especially for Netflix and Apple.

One final thing worth noticing is that linear regression seems to model the variations in stock price better, with pointier spikes and sharper direction changes. In contrast, the LSTM model predicts rounder spikes and smaller fluctuations. One cause for this could be the large batch size used while training the LSTM model, causing it to make average changes rather than exact changes to the weights of the model for each input.

Since the simple linear regression model is more accurate, this suggests that the relationship between stock closing prices and their historical closing prices is a simple relationship.

VII. PREDICTING FUTURE STOCK PRICES USING THE BEST MODEL

Using the prediction of the model as input for the model, we can predict future stock prices using the linear regression model. We predicted future stock prices for all 5 companies, using the closing prices from April 29th to May 5th as our initial closing prices. The predictions for the five companies can be seen in Figure 12, while the predictions for Apple, Amazon, and Google can be seen more clearly in Figures 13,14,15. The linear regression model predicts a slightly decrease in stock prices overall and show sightly fluctuations in stock prices. It is normal to see price fluctuations in a stock market; however, the decreasing trends in stock close prices result from outside events.

Apple, Amazon, and Google all fluctuate the same, showing that the technology market as a whole experiences similar

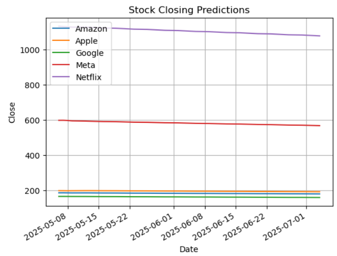

Fig. 12. Linear regression Closing Predictions

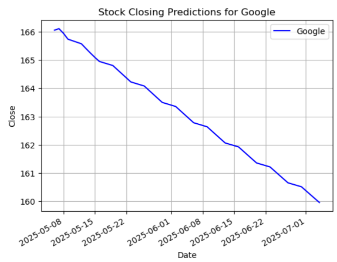

Fig. 13. Linear Regression Prediction On Google

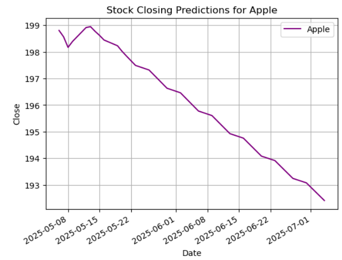

Fig. 14. Linear Regression Prediction On Apple

Fig. 15. Linear Regression Prediction On Amazon

changes in stock closing price. The difference width of the peaks and troughs, however, appear to be unique to the company.

As a link to reality, using the data that has already been affected by the 145% tariff announcement since April 12, 2025 would give us a total trend of a stock market price to decrease. Despite the technology market stocks decreasing, the technology market has ever growing importance. On April 29, General Secretary Xi Jinping emphasized during his inspection in Shanghai that Shanghai should enhance its function as a source of scientific and technological innovation and a leading role in high-end industries. This instruction further clarified the important position of scientific and technological innovation in the national development strategy, making some expectations for the development prospects of science and technology-related industries, and will make investors have a short-term optimism about the development prospects of science and technology-related industries around May 8th. We see this small period of stock price increase in the graph predicted by our model. However, as trade frictions between the United States and China intensify, the uncertainty of tariff policies makes it difficult for companies to make long-term plans, suppresses investment and consumption, which also makes investors gradually worry about the economic outlook, resulting in a continuous decline in stock closing prices.

We also generated predictions using the worse model (LSTM model). The predictions can be seen in Figure 16. Unlike the linear regression model, the LSTM model predicted an overall decrease in stock closing prices and failed to predict the normal fluctuations in stock closing prices. Although the model predicts total stock market collapse (very unlikely to actually happen) and thus performs badly, the ability to predict an overall trend (increase/decrease) is important. Perhaps another LSTM model with a slightly different structure could achieve both prediction of an overall trend and accuracy.

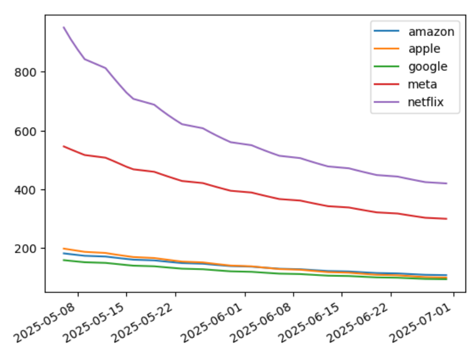

Fig. 16. LSTM Closing Predictions

VIII. A Model trained on weekly Data

We also trained an LSTM model on weekly data. The goal was to achieve a more generalized model that would predict overall increases or decreases in stock closing price rather than capturing daily fluctuations. The weekly data was gathered by taking the average over a week of daily data. Since stock market data is only collected during the traditional work week, our week consisted of five days. After creating the weeks, the data was handled the same as the data for the daily model, with one-hot encoding for the companies, a lag of seven weeks, and normalizing the closing prices to be between 0 and 1.

Like with the daily model, hyper-parameter tuning was performed via grid search on the weekly model. The model with the smallest MSE had parameters: Batch size: 32, Learning rate: 0.001, Dropout: 0.2, LSTM Hidden Size: 32, 1 linear layer, 1 LSTM layer, and a regularization parameter of 0.001 for Ridge Regularization (L2). One again, note that dropout was automatically set to 0 to avoid dropping input neurons since there is only one LSTM layer. A graph of Epochs vs MSE was used to visually determine that the optimal number of epochs is 10 epochs.

After initializing a model with the above hyper-parameters and training for 10 epochs, the weekly model achieved MSE: 0.001341, R^2: 0.9478, RMSE: 0.03661, and MAE: 0.02283. Compared to the models above for daily data, these scores reflect lower performance. The predictions on the testing data compared to the true prices are shown in Figure 17.

Fig. 17. Weekly Model Testing Predictions

In the graphs, it appears that the weekly model consistently under-predicts the true price. One reason for this could be that the model was trained on 80% of the data, meaning data from 2013 to about 2023. Due to inflation and technology advances, stock prices for each of the companies have increased since 2023. However, the model was not trained on data that reflected these increases, and thus may fail to predict them properly.

Similar to the daily model above, the weekly model also predicts total stock market collapse, at least in the technology sector (Figure 18). Thus, unlike our goal, the weekly model adds no more general information about stock market trends than the daily model.

We may find higher accuracy/more trends using a linear regression model on the weekly stock market data. However, due to lack of time, this was not fully explored during this project.

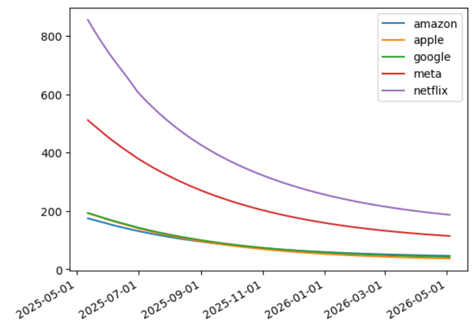

Fig. 18. LSTM Weekly Model Predictions

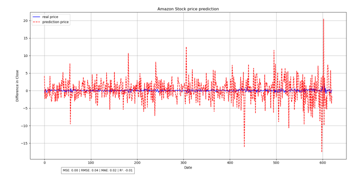

Fig. 19. Predictions on Difference in Closing Stock Price for Amazon

IX. Another Way of Normalizing the Data

For the above methods, the stock closing price was normalized via a MinMax Scaler, which sets values to be between 0 and 1 based on the maximum and minimum in the dataset. We also tried out another method of normalization, namely, rather than using stock Closing Prices, using the difference between stock closing prices. Each Difference was defined as today's price minus yesterday's price. Thus if the stock price increased, the difference was positive and if the stock price decreased, the difference was negative. Because the difference could theoretically be any real number, MinMax Scaler was applied to force the data to be between 0 and 1. The same seven day look back was then applied to the data before training the linear regression model. The goal with this method was to predict an overall increase or decrease in price rather than a specific price.

Unfortunately, this method of normalization did not lead to high performance. The model had MSE: 0.001467, RMSE: 0.03830, MAE: 0.02123, and R^2: -0.01251. Although MSE, RMSE, and MAE are fairly close to zero, R^2 is very far from 1 and is even negative, suggesting that the model does not account for the variance of the data. This is clear from the graph of Amazon's predicted differences in closing price in Figure 19 (the same is true for the other companies). Clearly, the variance is much larger than the true variance.

Because this model performed poorly, predictions for the future were not generated.

X. Future Work

The models above were trained only on previous days of stock data and company labels. However, the stock market is strongly affected by outside events, whether it be the Tariff war led by President Trump or the headlines regarding a particular company. Because of this, more accurate models could be created using outside information.

Additionally, the models were not given the information of exactly which date the stock closing price was taken from. It could be 7 days in 2014 or 7 days in 2023, and the model would not know the difference. Thus, we attempted to track the general trends in the stock market which remain unchanged over time. An interesting future project would be to include the datetime information while training the model to see what kind of past or future world events it predicts.

Finally, experimenting with different ways to normalize or process the stocks could lead to interesting results. For example, an earlier version of the LSTM normalized to be between -1 and 1. This model was found to be less accurate than the model with data normalized to be between 0 and 1. By creating models with different normalizations, we can explore which normalization stock data and particular models are best suited for.

Availability of Data and Materials

All models and EDA can be found on Github at the following link: https://github.com/YMoule/MAT-128-Project.git

References

[1] Emioma, Collins & S.O., Edeki. (2021). Stock price prediction using machine learning on least-squares linear regression basis. Journal of Physics: Conference Series. 1734. 012058. 10.1088/1742-6596/1734/1/012058.

[2] Gharehchopogh, F. S., Bonab, T. H., & Khaze, S. R. (2013). A linear regression approach to prediction of stock market trading volume: A case study. International Journal of Managing Value and Supply Chains, 4(3), 25–31. https://doi.org/10.5121/ijmvsc.2013.4303.

[3] Greg Hodd. (2023, April 8). Amazon Stock Forecasting in PyTorch with LSTM Neural Network (Time Series Forecasting) — Tutorial 3. Youtube. https://youtu.be/q_HS4s1L8UI?si=gBwlodL-cx9VaL65

[4] O P, S. & Karpagam, S. (2023). Multiple Companies Stock Price Analysis and Prediction Using LSTM. International Journal of Research and Analytical Reviews (IJRAR), 4(10). https://ijrpr.com/uploads/V4ISSUE10/IJRPR18528.pdf

[5] D. M. Q. Nelson, A. C. M. Pereira and R. A. de Oliveira, ”Stock market's price movement prediction with LSTM neural networks,” 2017 International Joint Conference on Neural Networks (IJCNN), Anchorage, AK, USA, 2017, pp. 1419-1426, doi: 10.1109/IJCNN.2017.7966019.

[6] Roy, S.S., Mittal, D., Basu, A., Abraham, A. (2015). Stock Market Forecasting Using LASSO Linear Regression Model. In: Abraham, A., Krömer, P., Snasel, V. (eds) Afro-European Conference for Industrial Advancement. Advances in Intelligent Systems and Computing, vol 334. Springer, Cham. https://doi.org/10.1007/978-3-319-13572-4_31

[7] ShankarLingam, G. & ThirupathiReddy K. (2025) Drawbacks of LSTM Algorithm: A Case Study. SSRN. https://papers.ssrn.com/sol3/Delivery.cfm/5080605.pdf?abstractid=5080605&mirid=1&type=2

[8] StatQuest with Josh Starmer. (2022, November 6). Long Short-Term Memory (LSTM), Clearly Explained [Video]. YouTube. https://www.youtube.com/watch?v=YCzL96nL7j0

[9] Y. Wang, S. Zhu and C. Li, ”Research on Multistep Time Series Prediction Based on LSTM,” 2019 3rd International Conference on Electronic Information Technology and Computer Engineering (EITCE), Xiamen, China, 2019, pp. 1155-1159, doi: 10.1109/EITCE47263.2019.9095044.