JOURNAL OF
EMERGING INVESTIGATORS

Article

Exploring the effects of diverse historical stock price data on the accuracy of stock price prediction models

Malaika Girma[1], Odysseas Drosis[2]

[1] International Community School, Addis Ababa, Ethiopia

[2] Project Instructor, Inspirit AI, California

SUMMARY

In the recent years, a majority of Americans have moved to investing in stocks through algorithmic trading. Not only does this underline the public's interest in the stock market, but it also shows the important role that artificial intelligence plays when it comes to predicting and trading stocks. These prediction models can benefit investors by improving their investment decisions and potentially increasing financial gains. In this research, we aimed to explore the impact of incorporating historical stock price data types, such as the opening prices, closing prices and highest prices, on the accuracy of stock price prediction models. The primary objective was to optimize the performance of our models. To carry out this research, we developed three supervised learning artificial intelligence models using Python: linear regression, neural network, and multiplicative weight update. Our models predicted the stock prices for Microsoft, Amazon, Google, Tesla, and Apple. First, the models used the opening prices for the past three days to predict the opening price on the fourth day. However, to enhance each model's performance, we evaluated whether adding extra information, such as closing and highest prices, would be beneficial. We hypothesized that incorporating the opening, closing, and highest prices would yield the highest accuracy as it would provide the models with the most information and help them better predict patterns in the stock market. The results supported our hypothesis as the models' average percent errors significantly decreased when they were given all three of these data types.

INTRODUCTION

The stock market is unpredictable and ever-changing due to various factors: unexpected events, global economic conditions, political events, and so forth. However, all these factors also present an opportunity to identify patterns in the stock market (1). As a result, the role of artificial intelligence (AI) in stock price predictions is rapidly increasing. AI is expected to significantly contribute to the projected market value of online trading, which is estimated to grow from $8.59 billion in 2021, to $12 billion by 2028 (2). There are many different strategies that a model can use to predict stock prices, including fundamental analysis, sentiment analysis and technical analysis. Fundamental analysis involves predicting a company's stock performance by analyzing data from the company's financial statements, while sentiment analysis involves incorporating news and social media information into the model (1). Technical analysis is a strategy that is useful for predicting stock prices for short-term investments. When using technical analysis, the model predicts a company's stock by examining data from the market, including historical returns, stock prices and the volume of historical trades (1). Typically, stock price prediction models utilize technical analysis as a foundation and supplement it with fundamental or sentiment analysis. This is because technical analysis provides a basic understanding for the model (3).

In our research project, we aimed to create three models that used technical analysis because we wanted to make short-term predictions using historical stock prices. To make these predictions, we made three models: linear regression, neural network and the multiplicative weight update (MWU). The linear regression model is a baseline model that takes the independent variables as input and uses weights to calculate the dependent variable (4). We used a linear regression model because we wanted to have a simple baseline model that would establish a standard level of performance for evaluating the other models and techniques. The neural network and MWU models are more complex. A neural network is a type of machine learning algorithm that mimics the neurons in the human brain as it has multiple connected nodes that work together to process and analyze data and help make predictions (4). We created this model because it can learn complex patterns and relationships in data. In addition, we used the MWU model which is a type of model that assigns weights to different inputs to help with its predictions. This MWU model was developed to serve as a combination model which consisted of the neural network and linear regression models as inputs. We developed this combination model because it is known to enhance the accuracy of predictions due to its ability to combine and assign weights to the predictions of multiple models. This means that it can give more importance to more relevant data which helps with its prediction accuracy.

When making our stock price predictions for Microsoft, Amazon, Google, Tesla, and Apple, we conducted three experiments to see whether inputting more information, such as the closing price and the highest price, in addition to the opening price, would improve our models' performances. The opening price is the price of the first stock that is traded when the market opens (5). Using the opening price for the three days is crucial as we predicted the opening price for the fourth day. The closing price is the price of the last stock that is traded when the market closes (5). The high price refers to the highest value that the stock reaches within that day (6). We hypothesized that incorporating the closing and highest prices, in addition to the opening price, would result in the

https://doi.org/10.59720/23-144

best predictions. We came to this hypothesis because we believed that the more information the models had, the more understanding they would have on the stock market, therefore allowing them to make better predictions. The results of our research supported our hypothesis as the models with all three data sets yielded the best results 80% of the time.

RESULTS

We performed three experiments to examine our hypothesis and determine if incorporating additional information would enhance each model's performance (neural network, linear regression, MWU) when predicting the opening stock price for five companies (Microsoft, Amazon, Google, Tesla, and Apple). The data consisted of the historical stock prices of each company within five years (2018-2022). In the first experiment, the models used opening prices from three consecutive days to predict the opening price on the fourth day. In the second experiment, the models used the opening prices and the closing prices from three consecutive days to predict the opening price of the fourth day. In the third experiment, the models used the opening prices, closing prices and highest prices from the last three days to predict the opening price on the fourth day.

Our hypothesis was that Experiment #3 would yield the best results because it would provide the models with the most information and improve their ability to predict patterns in the stock market. The accuracy of each of the three models was calculated separately for every company and experiment using the Average Percent Error (APE), where a lower APE indicates better model performance. To further evaluate the hypothesis, the combined APE was calculated for each company's predictions using the results from all three of the models (**Table 1**). This evaluated the overall performance of all three of the models for each company's stock prices.

Four out of the five companies (Microsoft, Google, Amazon and Tesla) showed the lowest combined APE in Experiment #3, compared to Experiment #1 and Experiment #2. This shows that the three models had the best combined performance during Experiment #3 **(Table 1)**. However, Apple had a better combined APE during the second experiment (**Table 1**). Although Apple's linear regression and MWU model had the best APE in Experiment #3, the APE for the neural network model worsened by 1.8% between Experiment #2 and Experiment #3 **(Figure 1)**. This caused Apple's combined

	Experiment Number		
	1	2	3
Microsoft	1.417	1.187	0.555
Google	1.587	1.049	0.988
Tesla	4.199	2.499	1.277
Apple	1.539	1.038	1.624
Amazon	3.885	2.899	1.328

Table 1. The combined Average Percent Error for all three models in each experiment. The green columns represent the experiment that performed the best for each company.

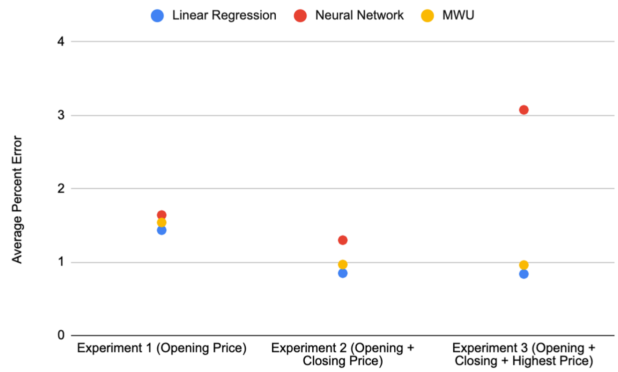

Figure 1: The Average Percent Error for all three models in all three experiments for Apple. Experiment #3 yielded the best results for two out of the three models.

APE for Experiment #3 to be worse than the combined APE for Experiment #2 (**Table 1**).

In addition to calculating the combined APE, we considered other variables as well. We noticed that 14/15 or 93.33% of the models improved between Experiment #1 and Experiment #2 **(Figures 1-5)**. All the models for Tesla, Microsoft, and Google (9 models total), improved between Experiment #2 and Experiment #3 (**Figures 2-5**). However, Experiment #3's outcome for Apple and Amazon were not as good. For Apple, two out of the three models improved between Experiment #2 and Experiment #3 (**Figure 1**). For Amazon, one out of the three models improved between Experiment #2 and Experiment #3. (**Figure 2**). This means that out of the six models for Amazon and Apple, three models improved while three models worsened. Therefore, in total, 12/15 of the models improved in their performance between Experiment #2 and Experiment #3.

Our hypothesis stated that the models would perform the best in Experiment #3 because they were fed the most information. Our experiments supported our hypothesis because Experiment #3 yielded the best results 12 out of the 15 times **(Figures 1-5).** In addition, 4 out of the 5 companies reached the lowest combined APE in Experiment #3 **(Table 1)**.

Neural Network Results

When it comes to the neural network design, the model was trained for 15 epochs, but due to the limited dataset, the model began to overfit, and this caused the models' APEs to reach about 70-90% on the testing set. To prevent overfitting, the epoch was reduced to 5 and this drastically improved the model's predictions, moving the APE to around 0.4%-8.487% on the testing set. It is important that even after this change, the neural network was often the worst performing model for all the companies and experiments which may suggest a need for more changes to the epochs.

DISCUSSION

Experiment #3 yielded the best results 80% of the time, which supports our hypothesis and shows that adding additional information usually improves the models' results. This could be because the additional information provides the models with a deeper understanding of the trends in the stock market. However, we noticed that the models improved more

JOURNAL OF
EMERGING INVESTIGATORS
https://doi.org/10.59720/23-144

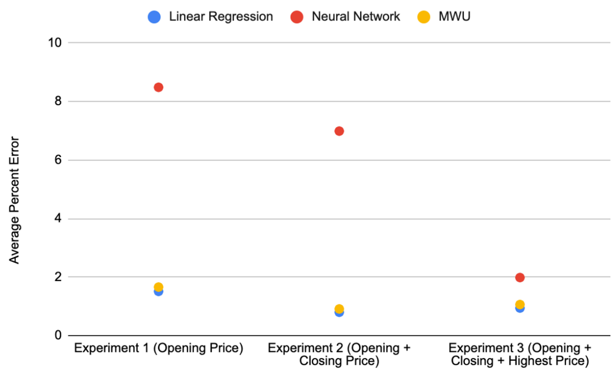

Figure 2: The Average Percent Error for all three models in all three experiments for Amazon. Experiment #3 yielded the best results for one out of the three models.

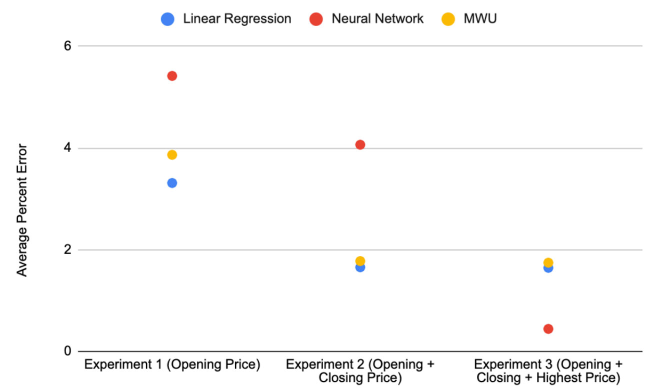

Figure 3: The Average Percent Error for all three models in all three experiments for Tesla. Experiment #3 yielded the best results for all three models.

between Experiments #1 and #2 (when we added the closing price) than between Experiments #2 and #3 (when we added the highest price) **(Figure 1-5)**. This may suggest that the closing prices provide the models with a greater comprehension of the stock market trends compared to the highest prices. This could be because closing prices represent the final traded price for a given trading day, indicating the market's value for that stock at the end of the session. In contrast, highest prices, while informative, solely reflect the peak values reached during the trading day and therefore may not capture the broader market trends.

One surprise was that the linear regression model often outperformed the neural network and MWU model. We weren't expecting this because the linear regression is a baseline model, whereas the neural network and MWU models are more advanced. However, it is possible that their poorer performance is due to overfitting. Overfitting occurs when the models become too specialized and start to retain excessively complex and intricate patterns that are specific to the training data but do not extend well to unseen data. This means that the models begin to memorize the data rather than learning the underlying concepts or relationships.

Unlike the linear regression and the MWU model, humans play a big role in the neural network's design meaning that we decided the size of the input layer and hidden layers for the neural network rather than the algorithm itself. The neural network's worse performance may be attributed to the chosen size of the layers. Modifying these parameters may potentially enhance the model's performance. This is a limitation because it doesn't show us the full extent of the model's prediction power. Another limitation could be that the combined APE may not be the best way to interpret the combined performance of the models. This is because one model's performance can affect the entire score for the combined APE. For example, when predicting the opening stock price for Amazon, the performance of the MWU model and the linear regression model worsened between Experiment #2 and Experiment #3. However, the neural network improved by a greater amount, causing the combined APE in Experiment #3 to be better than in Experiment #2 **(Figure 2).**

One example of a better metric to use (rather than the combined APE) is the Mean Bias Error (MBE) which measures the average difference between predicted and actual values, indicating whether the model tends to over-predict or under-predict. Knowing if a model over predicts or under predicts is beneficial as it can allow us to make changes to the models accordingly (7).

Our future plan is to enhance our models by integrating sentiment analysis through the inclusion of news articles and social media sentiment as input. Incorporating these diverse data types will enable the models to access a broader range of information, thus enhancing the potential for improved accuracy in predictions (8). We had intended to incorporate sentiment analysis in this research project; however, we were unable to find a suitable dataset containing news articles and social media comments. So far, we have a sentiment-analysis model that determines whether a piece of text has a positive or negative sentiment. It is important to note that the integration of news articles and social media data into the model may encounter challenges, such as the presence of unclean data and unrecognized characters, potentially impacting the performance of the sentiment analysis model. Therefore, we will instruct the model on how to handle these characters when it encounters them in the data.

Overall, this project supported our hypothesis because Experiment #3 yielded the highest prediction accuracy 80% of the time. This showed that giving a model more information can help improve its predictions because it will have a better understanding of the patterns within the stock market. This might be because using the closing prices provided insight into the daily price trend of the company. In addition, incorporating the highest price may have indicated the upper boundaries of the trend which gave our models a more precise representation of the stock trends. However, the results also indicated that some information may have greater predictive value than other information. We observed this when the models exhibited the most improvement with the inclusion of the closing price compared to the inclusion of the highest price.

MATERIALS AND METHODS

Data set / features

The data set was imported from Yahoo Finance and had five years of historical stock prices (year 2018-2022) for each of the five companies. The data was stored in a table where the X matrices held the inputs that were used to make the

JOURNAL OF
EMERGING INVESTIGATORS

https://doi.org/10.59720/23-144

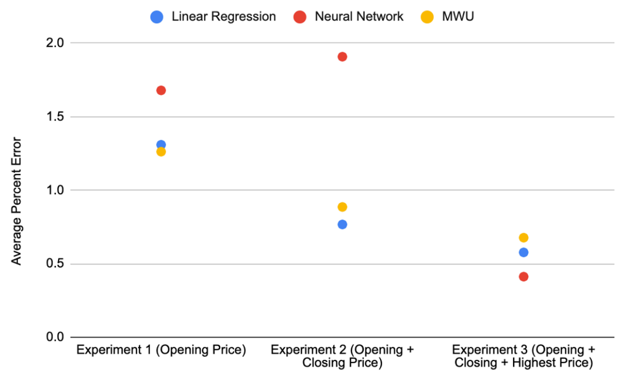

Figure 4: The Average Percent Error for all three models in all three experiments for Microsoft. Experiment #3 yielded the best results for all three.

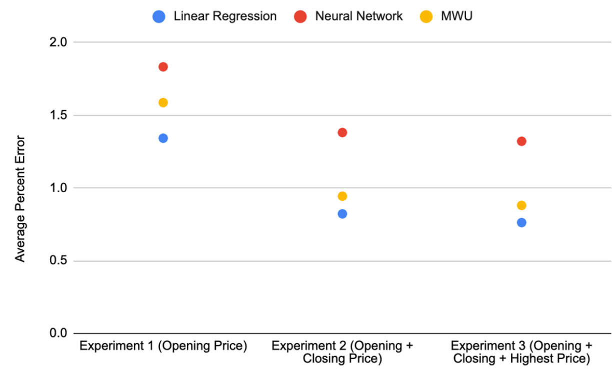

Figure 5: The Average Percent Error for all three models in all three experiments for Google. Experiment #3 yielded the best results for all three models.

prediction for the fourth day, which was then held in the Y matrices. All our three experiments required different amounts of data. Therefore, the size of the two-dimensional NumPy array was different in all three experiments. For the first experiment, the X matrices were two-dimensional NumPy Arrays with 1259 rows and 3 columns. The three columns of the X matrices held the three opening prices that were used to make the prediction for the fourth day. For the second experiment, the NumPy Array had 6 columns which held the three opening prices and the three closing prices. For the third experiment, the NumPy Array had 9 columns (1259 x 9) which held the three opening prices, the three closing prices and the three highest prices. The X matrices had 1259 rows due to their being 1259 available data points. All the elements of X were initialized to 0 using the np.zeros function. The Y matrices were 1-dimensional Python lists of length 1259. All the elements of Y were initialized to 0.

Splitting the Data

There was no need to process the data set or extract any features from it before splitting it. When we split the data, we used the train_test_split function from sklearn, which randomly put 67% of the data towards the training set and 33% of the data towards the testing set.

Building the linear regression model

Linear regression is a baseline model that takes the independent (x) variables as input and uses weights to calculate the dependent variable (y). During its training, our algorithm assigned a weight to each parameter in the dataset that reflected the importance of the parameter. In our case, the parameters were the prices from the three days used to make the prediction on the fourth day. Days with higher weights are given more importance during predictions, while days with lower weights are given less importance. The linear regression model followed the following equations for each experiment:

Experiment 1:

$$w_1o_1 + w_2o_2 + w_3o_3 = \text{prediction}$$

Experiment 2:

$$w_1o_1 + w_2o_2 + w_3o_3 + w_4c_1 + w_5c_2 + w_6c_3 = \text{prediction}$$

Experiment 3:

$$w_1o_1 + w_2o_2 + w_3o_3 + w_4c_1 + w_5c_2 + w_6c_3 + w_7h_1 + w_8h_2 + w_9h_3 = \text{prediction}$$

In this case, 'o' represents the opening price, 'c' represents the closing price, 'h' represents the highest price, 'w' represents the weight (which is different for each data point), 1 represents day 1, 2 represents day 2, and 3 represents day 3. The weights were learned using an optimization algorithm (which is a part of the linear regression model) and adjusted to minimize the difference between the predicted values and the actual values in the training set.

Building the neural network model

A neural network has three major sections: input layer, hidden layers and output layer (3). During our first experiment, our input was the three opening prices, so the input layer consisted of three nodes. During our second experiment, our input was the three opening prices and the three closing prices, so the input layer consisted of six nodes. During our third experiment, our input was the three opening prices, the three closing prices, and the three highest prices, so the input layer consisted of nine nodes. There were five hidden layers which had the following numbers of nodes for all three experiments: 20, 15, 10, 8, and 5. The output layer always had one node. The process of training a neural network involves adjusting the weights of the nodes in the hidden layers so that the network can accurately predict the output for a given input. This was done by using an optimization algorithm that adjusts the weights of the nodes to minimize the difference between the predicted and actual output in the training set. In a single epoch, the neural network examined every example in the dataset once and modified its weights to enhance its performance (3). Initially, we used 15 epochs, but then switched to 5 epochs due to overfitting.

Building the multiplicative weight update (MWU) model

The multiplicative weight update (MWU) model made predictions by using the linear regression and neural network model. The model's input consisted of both the test set and the predictions made by the two models on the test set. The model made its own predictions by assigning a weight to each

https://doi.org/10.59720/23-144

model based on the difference between their predictions and the actual value of the stock. The weights were then normalized to sum up to one and used to calculate a weighted average prediction for each example in the test set. This model did not train on any specific input data; instead, it provided a mechanism for updating weights based on the observed data.

Average Percent Error (APE) Calculation

When measuring the accuracy of the models, we used a statistical metric called the Average Percent Error (APE). This metric calculates the average percentage difference between our models' predicted stock values and the actual values in the market. A lower APE indicates a higher level of accuracy while a higher APE suggests greater deviations between the predicted and actual values. The combined APE was calculated using the following equation for each experiment: (linear regression APE + neural network APE + MWU APE)/3.

Received: June 29, 2023
Accepted: October 24, 2023
Published: September 24, 2024

REFERENCES

1. Li, Katherine (Yi). *"Predicting Stock Prices Using Machine Learning."* Neptune.Ai. neptune.ai/blog/predicting-stock-prices-using-machine-learning. 16 May 2023. Accessed 28 May 2023.
2. *"With 5.1% CAGR, Global Online Trading Platform Market Size Worth USD 12.16 Billion in 2028."* GlobeNewswire News Room, Fortune Business Insights.www.globenewswire.com/news-release/2022/06/14/2461792/0/en/With-5-1-CAGR-Global-Online-Trading-Platform-Market-Size-Worth-USD-12-16-Billion-in-2028.html#:~:text=This%20information%20is%20given%20by,5.1%25%20during%20the%20forecast%20period. 14 June 2022. Accessed June 30 2023.
3. Picasso, Andrea, et al. "*Technical Analysis and Sentiment Embeddings for Market Trend Prediction.*" *Expert Systems with Applications*, vol. 135, 2019, pp. 60–70, doi.org/10.1016/j.eswa.2019.06.014. Accessed May 4 2023.
4. R, S. (2021, August 16). "*A walk-through of regression analysis using artificial neural networks in tensorflow*". Analytics Vidhya. www.analyticsvidhya.com/blog/2021/08/a-walk-through-of-regression-analysis-using-artificial-neural-networks-in-tensorflow. Accessed July 29 2023.
5. Norris, Emily. "*Why Don't Stocks Begin Trading at the Previous Day's Closing Price?*" Investopedia. www.investopedia.com/ask/answers/139.asp#:~:text=The%20prices%20of%20stocks%20are%20fluid%20and%20constantly%20changing%3B%20the,time%20that%20stock%20was%20traded. 8 Sept. 2021. Accessed May 7 2023.
6. Chen, James. "*Today's High.*" Investopedia. www.investopedia.com/terms/t/todayshigh.asp#:~:text=Today's%20high%20is%20the%20highest%20price%20at%20which%20a%20stock,when%20calculating%20a%20moving%20average. 8 July 2022. Accessed July 15 2023.
7. Kolsoumi, S. (n.d.). *"Mean Bias error Calculator".* https://agrimetsoft.com/calculators/Mean%20Bias%20Error#google_vignette. Accessed July 15 2023.
8. Mehta, Pooja, et al. "*Harvesting Social Media Sentiment Analysis to Enhance Stock Market Prediction Using Deep Learning.*" *PeerJ Computer Science*, vol. 7, 2021, doi.org/10.7717/peerj-cs.476. Accessed May 4 2023.