Enhancing Stock Market Forecasting Using Machine Learning Models

Andrea Lim

Crimson Global Academy, Orlando, FL USA
Corresponding Author Email: *andrealim811[at]gmail.com*

Abstract: ***This study explores the integration of machine learning techniques in forecasting stock market trends. Focusing on three core models—linear regression, neural networks, and decision trees—it compares their predictive accuracy using historical data from leading technology firms. Additional enhancements such as sentiment analysis, time series extrapolation, and cross-company data incorporation are tested to evaluate their influence on prediction performance. Findings suggest that linear regression consistently outperforms other models in accuracy, with time series augmentation providing notable improvement. This work highlights the potential of hybrid AI-driven approaches to refine financial forecasting models.***

Keywords: Machine Learning, Stock Forecasting, Linear Regression, Sentiment Analysis, Time Series Modeling

1. Introduction

Since the launch of OpenAI's ChatGPT in November of 2022, there has been an explosion of AI technology: chatbots, search engines, speech recognition, and other AI-enabled applications. But AI has expanded beyond consumer goods, and is driving the profitability of the business world. Karlene C. Cousins, chair of FIU Business' Department of Information Systems and Business Analytics, says that "every business professional should have an understanding of how to use AI to add value to their business" (Florida International University, 2020). For efficiency, processing, and more, the finance industry has especially been implementing new technologies, leaning towards these methods for predicting the volatile stock market, believing that it will transform the global economy. Predicting the stock market is notoriously challenging, as there are so many variables and factors to consider, some being impossible to predict such as geopolitical events and natural disasters. Stock price predictions help both consumers and producers, as both sides can gain significant profits, and provide investors and financial analysts with the information needed to make decisions concerning issues that can result in cutting losses. There have been numerous developments for more and more accurate models that provide insights into the future of the stock market. Cornell researchers have built a machine learning model that uses machine learning and data from financial news to predict financial returns much more accurately than the standard, traditional model (DiPietro, 2023). This development, a significant advancement in the finance industry thanks to artificial intelligence, is one of the many examples of AI's potential in revolutionizing the world. This study aims to compare the performance of machine learning models in predicting stock market trends and assess how external features such as sentiment data and cross-company metrics influence predictive accuracy. This research is significant as it offers insights into how machine learning models can enhance financial forecasting, a critical component in strategic investment decisions and economic planning.

Four machine learning models will be used in this study—linear regression, neural network, decision tree, and random forest. A linear regression model uses the linear relationship of two variables, the independent and dependent variables, to find the line that best fits the data points to minimize residuals, the distance from the prediction and actual stock value (Analytics Vidhya, 2023). A neural network model consists of layers of nodes that are, in turn, connected to several other nodes in the next layer. Each node is first assigned a random number as a weight, and after the model's prediction is compared to the actual answer, the weights are adjusted accordingly until the predictions are similar to the answer. Similar to neural networks, the decision tree model consists of decision nodes that branch off into more nodes that depict decisions to be made. A random forest model uses several decision trees to make its predictions, making it more accurate than a decision tree but much more complex (Wohlwend, 2023). Three of these models (linear regression, neural network, and decision tree) will be analyzed and compared to each other using varying factors for predicting, such as incorporating the open price of Microsoft in predicting Google's prices, and a basic simulation of a stock trade was built at the end to test the performance of the model. This study aims to test these models' capabilities and compare the different factors and models used.

2. Materials and Methods

2.1 Splitting the Datasets

Using the yfinance library, Google's financial data over the last five years was imported. Unnecessary information, 'High', 'Close', 'Volume', 'Dividends', 'Stock Splits', were discarded from the dataset, leaving only the open price to be turned into a list. Since the machine learning models would examine the previous four days to predict the price of the next day, a variable named "X" was created to store every four consecutive open prices while a variable named "Y" would

Paper ID: SR25730031803 DOI: https://dx.doi.org/10.21275/SR25730031803

store X's corresponding open price of the fifth day. In simpler terms, X stores the previous four days so that the machine learning models can be trained to predict the fifth day, or the next day, which is what Y holds. The data was split into training and testing sets named X_train, X_test, Y_train, and Y_test. 33% (⅓) of the data was allocated for the testing sets while the remaining 67% (⅔) of the data was used to train the models. This method is called the Train Test Split, and it involves the training of the model with a training set, then using the testing set to compare the predictions of the model to the actual data. It is crucial for testing machine learning models as it provides an estimate on the performance of the model on future datasets.

2.2 Creating the models

Four models—linear regression, neural network, decision tree, random forest models—were created, along with the imports of their libraries, and were all trained with the training sets X_train and Y_train. The linear regression model was created first, named "regr", which was then trained using X_train and Y_train. It was tested using X_test, its predictions of X_test stored in a new list named Y_predict. The same method was applied on the other three models. When creating the neural network model, the number of its maximum iterations was set to 500, meaning that for the training process of the neural network model, it will repeat its cycle of comparing the output to the answer and adjusting its weights at a maximum of 500 times. The random forest model had its maximum depth, or maximum number of splits each decision tree can make, set to 10. Each model's prediction results from X_test were iterated through and compared to the answer from the dataset Y_test, the target variable. The average percent error was calculated by dividing the difference of the model's prediction and the true stock price it was attempting to predict by the true stock price. In each iteration, the percent error for each day was calculated by dividing the difference of the model's prediction and the data from Y_test by Y_test. The result was then added to a variable that would later be divided by the length of Y_test, calculating the average percent error. To account for the "blanks" in the data that were replaced with 0, messing up calculations, the code was rewritten so that it would skip the day if the data was less than 5. Five of the major corporations (Google, Apple, Microsoft, Meta, Amazon) were used to test the models. Results summarized in Table 1 are further interpreted in the Discussion section.

2.3 The Ensemble Method

To further compare the three models, the Ensemble Method was used to combine these models to produce one final, precise prediction. Considering the fact that the accuracy of each model is dependent on the type of data given, their weights were taken into account so that the model with the most accuracy would contribute most to the final prediction while the model with the least accuracy wouldn't affect the answer as much. Similar to when calculating each model's percent error, the day was skipped if the data value was less than 1. Each model was first assigned a weight equal to 1, then the prediction, using all three weights, was calculated by dividing the sum of each model's prediction multiplied by its corresponding weight divided by the sum of all three weights. Variables, corresponding to each model, named error1, error2, and error3, stored the percent error of the models' performances by finding the difference between each model's prediction and the actual answer from the dataset Y_test and dividing it by Y_test. Each model's weight was updated by dividing its current value by 2 raised to the power of its percent error that was previously calculated. The final updated weights were equal to its current weight divided by the sum of the values of the weights, and this process continued until the entire dataset was iterated through.

2.4 Using Different Companies to Predict Google's Stocks

To investigate the effect of one company's stocks on another, Google's stock market performance was evaluated with the open price data of other companies (Microsoft, META, Apple, Amazon) as one of the factors for training the models. Microsoft's open price data from the last five years were imported, and the variable X, which used to hold every four consecutive open prices, was altered so that it would also take the open price of the previous day of Microsoft's data into account, therefore having five open prices data. The same was done for other companies, and when using both Microsoft and META as factors, X was enlarged to hold six values and the previous day of META's data was included along with Microsoft's. Results of each model's average percent error and weight are shown in Table 2.

2.5 Simulation

A basic stock trade was predicted using the models predictions made by the linear regression model. A variable named budget was created, initialized to 10000, and a variable named initial_stock was set to 0. The trained linear regression model was used to make predictions on the dataset X. The open prices are then iterated through, and the code checks if the current price is lower than the model's prediction. If the budget is greater than or equal to the day's stock price and the next day's predictions are also greater than the day's stock price, then 1 would be added to the initial_stock variable and the open price value would be subtracted from the budget. If the initial_stock variable is greater than or equal to 1 and the next day's prediction is less than the day's open price, then 1 would be subtracted from the initial_stock variable and the day's open price would be added to the budget. In summary, the simulation buys the stock if there is a growth in price the next day and sells the stock if there is a decrease in the price. When the code was done iterating, the product of the initial_stock and the last day's price was added to the budget and initial_stock was set to 0, simulating the selling of any remaining stocks. The starting and final budgets were recorded and the percent increase was calculated, as seen in Table 3.

2.6 Sentiment Analysis

An additional feature, sentiment analysis, was added to the model to predict Google's stocks based on its financial reports. From Hugging Face, a company that has tools for building applications using machine learning, a pre-trained model in financial news sentiment analysis was imported, called "distilroberta finetuned financial news sentiment analysis". From the transformers package, used for Natural Language Processing (NLP) models, the necessary classes and packages, such as PyTorch and AutoTokenizer, were imported. The model and tokenizer, which transforms text data so that the model can process it, were loaded. Because the open price data of the companies from over the last five years were previously used, Google's financial news articles from 2019 to 2024 were found. With the finance industry's use of fiscal quarters, Q2 and Q4 reports were used each year with the exception of 2024 having only its Q1 report. One to two sentences, describing how Google's revenues were during that quarter, were taken from each of the eleven articles and stored in a list. The list was iterated through and "tokenized" using the previously loaded tokenizer tool. The tokenized texts were run through the model and its results, numbers corresponding to the sentiment (0 for negative, 1 for neutral, and 2 for positive), were outputted then stored in a list called "z". Human evaluation was used to manually check the numbers to make sure the model's output makes sense. X, the variable that stored Google's open prices from the past five years, was separated into eleven segments, and each segment was iterated through and had their last item replaced by the corresponding item from the z list. For example, the last element of the first segment was replaced by the first number in the sentiment score list, while the last element of the second segment was replaced by the second number in the z list. The code was run and the results of the models' performances were printed out, displayed in Table 4. Another Hugging Face pre-trained model was uploaded, called "google play sentiment analysis 300k", which followed the same process as the previous model.

2.7 Time Series

An approach to time series was made by assuming that the models' predictions are true, then extrapolating to predict the next stock prices. X was changed to hold the first element of the first five open prices, so that it would hold integers rather than arrays. Y was changed to hold the open price of the sixth day. The open prices were looped through, and for each day, a variable named predict was set to the linear regression model's prediction of Y using X's five open prices. Once predict was found, X's elements were all shifted to the previous index. For example, the second element replaced the first element, the third element was moved to the second element's spot, and so on until the fifth element, which was replaced by predict, the linear model's prediction of the next day. The average percent error was calculated by dividing the difference between the prediction and Y by Y.

3. Results

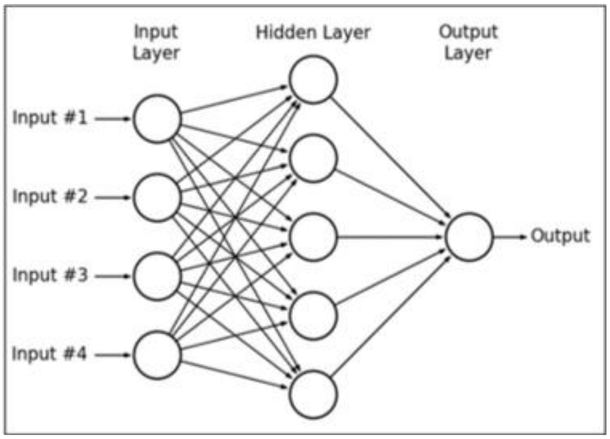

Figure 1: Structure of a Neural Network Model

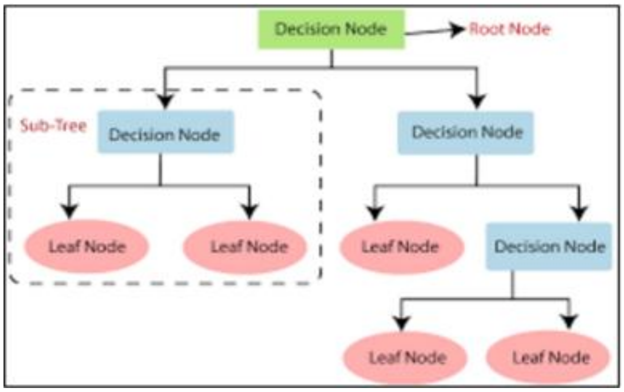

Figure 2: Structure of a Decision Tree Model

Table 1: The Models' Average Percent Error and Weight for Predicting the Stocks of the Companies

Company	Linear Regression		Neural Network		Decision Tree		Time Series
Google	1.33%	0.57	1.33%	0.34	2.20%	0.09	0.80%
Apple	1.29%	0.53	1.29%	0.41	2.03%	0.06	0.87%
Microsoft	1.21%	0.53	1.21%	0.37	1.78%	0.10	0.92%
Meta / Facebook	1.75%	0.59	1.75%	0.36	2.60%	0.05	0.82%
Amazon	1.60%	0.50	1.60%	0.43	2.28%	0.07	0.99%

Table 2: The Effects of the Open Price of Another Company on Google

Company	Linear Regression		Neural Network		Decision Tree	
Apple	1.48%	0.50	1.48%	0.42	2.12%	0.08
Microsoft	1.31%	0.66	1.31%	0.25	2.02%	0.08
META / Facebook	1.40%	0.60	1.40%	0.30	2.03%	0.10
Amazon	1.45%	0.55	1.45%	0.34	2.02%	0.11
Microsoft and META	1.38%	0.68	1.38%	0.17	1.90%	0.15

Table 3: Comparison of Results of the Stock Trade Simulation from Different Starting Budgets

Starting Budget	Final Budget with **Linear Regression**	Final Budget with **Sentiment Analysis**	Final Budget with **Time Series**
$100	$601.58	$2,253.99	$23,746,071.63
$1,000	$18,619.97	$4,701.11	$25,135,150.28
$10,000	$22,363.62	$20,596.18	$26,013,999.87
$100,000	$125,706.44	$111,657.07	$28,366,667.47
$1,000,000	$1,025,706.44	$1,011,657.07	$30,173,204.36

Table 4: Results of the Models' Performances with Sentiment Analysis on Google's Stocks

Model	Linear Regression		Neural Network		Decision Tree	
Model 1	1.89%	0.65	1.89%	0.31	2.80%	0.05
Model 2	1.87%	0.58	1.87%	0.35	2.62%	0.07

4. Discussion

Out of the three models that were compared to each other (linear regression, neural network, and decision tree), both the linear regression and neural network models had the same average percent error of the testing data. For the company Google's dataset, it was 1.33% while the decision tree model's average percent error was 2.20% for the same testing data for Google. This means that compared to the other two models, the decision tree model's performance for stock price prediction is not as suitable. To further evaluate the models' performances, the results of their weights, which were done so that each model can give more or less contributions to the final answer depending on its accuracy, were taken into account. For every company, the linear regression model had the heaviest weight, shown in Table 1, which shows both the average percent error and weight of the model for each company. For Google's dataset, it had a weight of 0.57, the neural network weight was 0.34, and the weight of the decision tree model was 0.09. These numbers are able to approximately show the performance level of the models. Both the linear regression and neural network models are undoubtedly more accurate than the decision tree model, shown by how their weights were about four times greater than the decision tree model's. The linear regression, consistently having the heaviest weight for each company's data set, seems to be the most accurate predictor. The results can be explained by the way the models work.

Regression models a target value based on independent predictors and is therefore commonly used for determining cause and effect relationships between variables (Gandhi, 2018). A linear regression model shows the linear relationship between two variables, the independent and dependent variable, and finds the line that best fits the data points on a graph so that its residuals, the distance from the prediction and the actual value, are minimized ("All You Need to Know," 2023). Therefore, linear regression models are best used for predictions and regressions. A neural network model consists of layers of nodes, each individual node connected to several other nodes in the next layer. Each node is first assigned a random number as a weight, and after the model's prediction is compared to the actual answer, the weights are adjusted accordingly until the predictions are similar to the answer. Similar to neural networks, the decision tree model consists of decision nodes that branch off into more nodes that depict decisions to be made. Both neural network and decision tree models are best for identifying important variables and the relationships between them. A random forest model uses several decision trees to make its predictions.

Therefore, the decision tree model was the least performative due to how it works—it's best for making decisions rather than predictions, while the linear regression model was the most accurate predictor because it was built for that purpose.

Taking the open price of another company into account of Google's open prices resulted in varying changes of the average percent error. With Microsoft's data taken into account for predicting Google's stocks, the average percent error of the linear regression and neural network models decreased from 1.33% to 1.31%. Similarly, the decision tree model resulted in a decrease from 2.20% to 2.02%. The effects of the other companies—Apple, META, Amazon—as well as both Microsoft and META on Google's stocks, were tested. All of these testings showed an increase in the average percent error, shown in Table 2. Taking two companies, Microsoft and META, into account was expected to show a decrease in the average percent error for all three models, but an increase from 1.33% to 1.38% was observed in the linear regression and neural network models and a decrease from 2.20% to 1.90% for the decision tree model. It can be concluded that the open prices of Microsoft and Google have a correlation, and while taking into account two companies does provide more data for the machine learning model to train with, it does not guarantee an increase in accuracy.

For the simulation of a basic stock trade, the linear regression model proved to be successful as our percent increase of the budget was 123.64%, starting from $10,000 and ending the simulation with $22,363.62. This simulation was tested by having it start with different budgets from $100 to $1,000,000, and found that the peak of the budget's percent increase is when the starting budget is approximately $1,000 with a percent increase of 17612.00% that leaves the budget at $18,619.97. With the sentiment analysis model included, there was a variation in its More information is shown in Table 3.

With the sentiment of financial news taken into account, the models had a slightly greater percent error. The Linear Regression and neural network models had a 1.89% error and the decision tree model's was 2.80%. However, the linear model's weight increased from 0.57 to 0.64, the neural network model decreased from 0.36 to 0.31, and the decision tree's model went from 0.07 to 0.05. Although the percent errors all slightly increased, the sentiment analysis scores do matter. The linear model, as it's best for prediction, is able to distinguish whether the sentiment analysis was effective or not. Seeing that the linear model has more weight and the fact that none of the models' weights is 0, the numbers from the financial news reports must carry some weight for the model's prediction. While the sentiment analysis does affect the model, only taking eleven reports into account do not have the power to improve its performance. Reports taken on an everyday basis would most likely have a greater effect on the models and significantly improve its prediction accuracy. Although the second sentiment analysis model claimed to have an accuracy of 0.5654, compared to the previous model's accuracy of 0.9823, it did slightly better than the first. With the second model taken into account, the percent errors of the three models were lower, as the linear regression and neural network models had a 1.87% error and the decision tree model had a 2.62% error. While the weight of the linear regression model decreased to 0.58, the weights of the neural network model increased to 0.35 and the decision tree's increased to 0.07.

An approach to time series was made by assuming that the models' predictions are true. Time series is a dataset that tracks a sample of consecutive data points over time. Time series analysis is used to see how a given variable changes over time and what factors influence certain variables from period to period.

These findings of a successful linear regression model can be used to aid investors and financial analysts in creating strategies for maximizing profits and saving money. Consumers, who are at risk due to the volatility of the stock market, can use the model to decrease their losses. More research could be done on more factors that play a role in the stock market, such as economic indicators such as the GDP. Additionally, testing more complex approaches to sentiment analysis on financial news, on an everyday basis rather than twice a year, could potentially result in much more accuracy. This strategy of taking online text, whether from company reports or social media posts, and extracting the positive or negative words associated with the company, have been used by investors to predict a stock price's rise or fall (DiPetro, 2023), and Cornell researchers have combined AI and online news to produce a highly accurate model.

References

[1] Analytics Vidhya. (2023, July 20). *All you need to know about your first Machine Learning model – Linear Regression*. https://www.analyticsvidhya.com/blog/2021/05/all-you-need-to-know-about-your-first-machine-learning-model-line ar-regression/#h-understanding-linear-regression

[2] Claessens, S., & Kose, A. M. (n.d.). *Recession: When bad times prevail*. International Monetary Fund. https://www.imf.org/external/pubs/ft/fandd/basics/recess.htm

[3] DiPietro, L. (2023, July 11). *Data scientists predict stock returns with AI and online news*. Cornell Chronicle. https://news.cornell.edu/stories/2023/07/data-scientists-predict-stock-returns-ai-and-online-news

[4] Gandhi, R. (2018, May 27). *Introduction to machine learning algorithms: Linear regression*. Medium. https://towardsdatascience.com/introduction-to-machine-learning-algorithms-linear-regression-14c4e325882a

[5] Hardesty, L. (2017, April 14). *Explained: Neural networks*. MIT News. https://news.mit.edu/2017/explained-neural-networks-deep-learning-0414

[6] Li, J., Liu, L., & Green, R. (2017, October 10). Building diversified multiple trees for classification in high dimensional noisy biomedical data. *BMC Bioinformatics*, *18*(1), 1–14. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5634968/

[7] Florida International University. (2020). *The AI revolution: How artificial intelligence is driving better business decision-making*. https://business.fiu.edu/magazine/fall2020/the-ai-revolution.html

[8] Tuovila, A. (2024, February 26). *Forecasting: What it is, how it's used in business and investing*. Investopedia. https://www.investopedia.com/terms/f/forecasting.asp#toc-the-bottom-line

[9] Master's in Data Science. (n.d.). *What is a decision tree?* https://www.mastersindatascience.org/learning/machine-learning-algorithms/decision-tree/

[10] Wohlwend, B. (2023, July 23). *Decision tree, random forest, and XGBoost: An exploration into the heart of machine learning*. Medium. https://medium.com/@brandon93.w/decision-tree-random-forest-and-xgboost-an-exploration-into-the-heart-of-mach ine-learning-90dc212f4948