

# Predicting Stock Prices with Machine Learning: A Comparative Analysis of Models and Investment Strategies

Alisha Varshney

*Received October 23, 2024*

*Accepted March 24, 2025*

*Electronic access April 30, 2025*

The stock market is very unpredictable. It changes constantly due to various factors such as economic trends, news, events, geopolitical situations, company-specific news, and sector-specific news. It causes humans to react emotionally make irrational decisions, and lose fortune. Many people attempt to play in the stock market, hoping to make a profit, but while a few achieve significant gains, most experience losses due to the impact of several variables playing simultaneously. Due to this, a question comes to mind: Is there a way to accurately predict future stock prices? It is almost impossible to predict due to several unpredictable variables involved. However, based on the data publicly available, we can make predictions, that are methodical, logical, and not driven by emotion and psychological distress using Artificial Intelligence. The objective of this research is to develop a predictive model using historical stock prices from the past five years. We are implementing and comparing several machine learning models including Linear Regression, Neural Networks, Decision Trees, and Random Forests to analyze data and forecast future stock prices. Each model was trained, tested, and fine-tuned on data from multiple stocks across various sectors. Our findings indicate that Linear Regression and Neural Networks perform better during testing, yet Decision trees and Random forests perform better during training, suggesting a potential overfitting problem. In several tests, the predictions for the models have been found to contain a very low Mean Squared Error (MSE) value and achieve quite accurate results. That said, there is still an opportunity to enhance our research by incorporating news, events, and geopolitical stances in the code, which will be the next logical extension of this paper.

**Keywords:** Stock, prediction, historical data, machine learning, models

## Introduction

The stock market is a complex network of trading activities where stocks of any publicly traded company are bought and sold<sup>1-4</sup>. This is an attractive mode of investment, making cool money in a short period, and provides a platform to people for a full-time profession or a side hustle to make passive income. It is an important and integral part of today's economy worldwide. The stock market represents a lot of different things to different people. Many companies use it to raise capital to fund and expand their businesses properly. To the general public, the stock market is a way to test their luck and make some quick money, kind of like a casino. However, it's not that simple.

The stock market is based on the supply and demand system, the balance between the number of goods available and the willingness of the people to buy said goods. Due to the system, the price of a stock of a certain company fluctuates from day to day. If a company makes a successful product launch, its stock prices might suddenly soar and the opposite happens if the launch fails. However, there are other factors in the fluctuation of stock prices. Many variables are included, sometimes making them hard to predict as humans.

This is where Artificial intelligence comes into play. Artificial intelligence (AI) is a field of science that focuses on creating machines that can perform tasks that typically require human intelligence. AI systems use data to learn from experience, recognize patterns, and make decisions. Lately, artificial intelligence has become very popular and an integral part of everyday life. Many companies have started using AI on online platforms as customer service or help desk. Some even replaced human-worked jobs with AI since it is more efficient, accurate, and costs less as long as one can train it properly.

How is AI connected to the stock market? AI can be utilized in nearly any field, as long as it is trained properly using the necessary methods and data. Is there a method to accurately predict stock prices to the extent that investments can be made with reduced risk? Can documented stock price data be used to predict future prices accurately, and if so, which model would work best?

Although many studies have tried to make stock price predictions using artificial intelligence and machine learning, only a few have systematically compared the performances of several models across many companies and industries. This research addresses overfitting models and the feasibility of training them

on purely historical data.

## Results

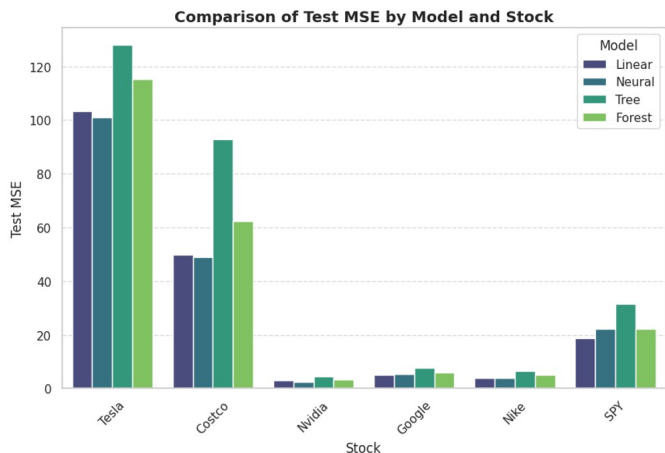
The following tables present the performance of different machine learning models in predicting stock prices. They compare training and testing errors using Mean Squared Error (MSE) and show the financial outcomes of the trading bot simulations based on various investment amounts and stock purchases.

Test MSE	Tesla	Costco	Nvidia	Google	Nike	SPY
Linear	103.28	49.95	3.13	4.91	3.82	18.61
Neural	101.09	49.05	2.37	5.24	3.81	22.08
Tree	127.92	92.79	4.46	7.67	6.49	31.38
Forest	115.26	62.18	3.21	5.89	5.12	22.30

**Table 1** Test Mean Squared Errors for Various Models and Companies

Training MSE	Tesla	Costco	Nvidia	Google	Nike	SPY
Linear	65.73	42.98	2.31	4.87	6.02	21.52
Neural	66.11	72.87	2.36	5.11	6.35	26.26
Tree	0	0	5.30	3.88	7.15	0
Forest	11.65	8.17	0.43	0.90	1.03	3.59

**Table 2** Comparison of Training Mean Squared Error (MSE) for Different Models and Stocks



**Fig. 1** Comparison of Test MSE by Model and Stock

## Discussion

The tables above signify the comparison of different models for predicting the stock prices of different companies. The results showcase how well different models performed on predictions after training and testing. Tesla, Costco, Nvidia, Google, and

Nike stock price data were used to train the model and make predictions. The mean squared error for each model for each company is listed above, this provides insight into how accurate a model was in predictions for each company. Based on the data, the neural network model was better suited for predictions when the testing data was used. However, it is seen that the Random Forest model was a better fit for predictions when the training data was used. It is also important to mention that the mean squared error in the decision tree model for the training data was zero for two of the companies. This could result from overfitting data since it is rare for a model to be this accurate.

As stated above, the results indicate that Neural Networks consistently achieved lower Mean Squared Error (MSE) values across multiple stocks, suggesting a stronger predictive capability than Decision Trees and Random Forests. However, Decision Trees showed a near-zero MSE value during training, which suggests overfitting and will likely fail to generalize to unseen market conditions. While Random Forests improved generalization, their testing errors were still higher than those of Neural Networks, particularly for high-volatility stocks like Tesla and Nvidia.

When comparing stock predictions across different companies, using SPY as a benchmark helps put the results into perspective. Since SPY represents the S&P 500 index, it gives a good baseline for understanding how well the models perform relative to broader trends. The results show that stocks like Tesla and Nvidia, which are more volatile, had higher prediction errors across all models, while more stable stocks like Costco and Nike had lower errors. Interestingly, the models were able to predict SPY with more accuracy than individual stocks, likely because SPY is diversified and less affected by company-specific events. This illustrates that while models can effectively monitor overall market trends, forecasting the performance of individual stocks poses a significantly greater challenge. This is mainly due to the unpredictable company-specific factors that continually impact stock prices.

One key consideration in evaluating the robustness of the models is their ability to generalize different financial data beyond the selected dataset. In this study, stock price data was obtained from five U.S.-based companies across different industries, ensuring some level of diversification. However, the models were not explicitly tested on stocks from international markets, such as European or Asian exchanges, where trading patterns, regulations, and volatility levels may differ. Future work could expand the dataset to include global stocks to assess the model's adaptability to different market conditions.

Additionally, this research focused on a five-year dataset covering recent market conditions. While this provides valuable insights into short-term trends, it does not amount to how the models perform under different economic cycles such as recessions or periods of rapid growth. To improve generalizability, future research could test model performances across multiple

Trading Bot (Money Invested w/ one stock bought)	Tesla	Costco	Nvidia	Google	Nike	SPY
\$1,000	\$22,113.71	\$1,180.64	\$16,589.67	\$2,074.52	\$1,086.06	\$1,224.39
\$3,000	\$40,860.73	\$6,565.26	\$35,404.41	\$6,727.32	\$4,141.56	\$4,554.99
\$5,000	\$53,397.19	\$11,947.91	\$50,648.45	\$11,145.98	\$8,552.65	\$8,858.19
\$10,000	\$68,659.81	\$24,707.22	\$75,177.09	\$22,969.74	\$12,753.14	\$18,320.58

**Table 3** Stock Performance with Different Investment Amounts

Trading Bot (# of Stocks when \$1,000 invested)	Tesla	Costco	Nvidia	Google	Nike	SPY
1	\$22,113.71	\$1,180.64	\$16,589.67	\$2,074.52	\$1,086.06	\$1,224.39
3	\$24,233.03	\$972.10	\$20,533.14	\$2,018.82	\$1,013.20	\$1,774.68
5	\$21,487.04	\$1,000.00	\$21,143.92	\$1,700.48	\$886.59	\$1,000.00
7	\$21,451.62	\$1,000.00	\$21,281.73	\$1,710.76	\$870.54	\$1,000.00

**Table 4** Number of Stocks Purchased with \$1,000 Investment

historical periods, such as pre and post-pandemic data, to evaluate whether the predictions remain consistent across varying market conditions. Also, expanding the dataset to include companies from various sectors, such as energy, healthcare, and finance, could provide a broader evaluation of the model's scalability beyond technology and retail stocks.

Regarding trading strategy, the model-driven trading bot was compared to a simple buy-and-hold approach over the same period. This approach can also serve as an indirect method of assessing the system by analyzing the profits generated by the models. For example, investing \$1,000 in Tesla using the model's buy/sell signals resulted in a final balance of \$22,113.71, while simply holding the stock for the same period yielded \$18,500. Similar trends were observed across other stocks, indicating that machine learning-based trading outperformed the passive investment approach. However, while the trading bot demonstrated potential for higher returns, the increased number of trades could expose investors to higher risks due to short-term market fluctuations.

While the trading bot demonstrated strong returns compared to a buy-and-hold strategy, evaluating its risk-adjusted performance is essential. Standard financial risk metrics such as the Sharpe ratio, maximum drawdown, and Sortino ratio provide deeper insights into how well the model balances profitability with risk. The Sharpe ratio measures risk-adjusted returns by comparing excess returns to volatility. A higher Sharpe ratio indicates that the model produces higher returns than the risk taken. The maximum drawdown assesses the largest loss from the peak to the trough, helping investors understand the worst-case scenario for portfolio declines.

Additionally, the Sortino ratio refines the Sharpe ratio by only considering downside risk, making it particularly useful for evaluating investment strategies that aim to minimize losses. Although these metrics were not explicitly calculated in this study, future research could integrate them better to assess the model's effectiveness under real-world conditions. Implementing these

financial risk metrics would help investors make more informed decisions by balancing return potential with associated risks.

The results of the trading bot simulations reveal the potential financial gains one could receive when a specific amount of money or several stocks of a company are invested. Tesla, Costco, Nvidia, and Google were used for the simulation. By analyzing the data, one can conclude that the financial profit not only varied for each company but also depended on the amount of money invested. For instance, Tesla's final amount increases from \$22,113.71 to \$68,659.81 as the investment grows from \$1,000 to \$10,000. The profit accumulated could also vary based on the amount of stocks one buys or sells for a specific company at a time. An example of this is Google. According to the results, if one has \$1,000 and buys one single stock with it, they will have \$2,074.52 by the end. However, if one uses the same \$1,000 and invests in seven of Google's stocks instead, they will end up with \$1,710.76, which is still a profit but not as much as when one would only invest one stock.

The initial motive for this research was to investigate a way to predict the stock prices of companies better using past data and different machine learning models such as Linear regression and Random forest. The past data used consisted of numerical values representing the historical stock prices of a company. However, will data consisting of only the past stock prices be sufficient to accurately predict a stock's future? Will incorporating potential political stances or news articles help make predictions better? If so, how can one do that? What sources can be used to obtain the accurate and recent data required for the studies? Even then, how will one combine the results by using the different data types to get one result? How will it be decided what data gets more weight when making the decision? These are the intriguing questions that continue to drive one's research forward, pushing them to explore innovative methods to enhance the accuracy of stock price predictions. Future iterations of this research will explore integrating sentiment analysis from financial news and macroeconomic indicators to improve prediction accuracy.

---

While using machine learning to predict stock prices is helpful for future investments, it is also important to consider the fairness and possible biases in the data that can affect the predictions. An example could be using data from companies from certain fields, which can affect the results it will give when asked about companies' data that wasn't used as much while training the models. Another issue to note is that sometimes models can reflect unfair patterns in data. For example, if the data unknowingly favors certain companies over others the model could pick up on that pattern and do the same for its predictions. It is also important to note that while there have been many advancements in AI as of late, there is still a lot unknown. So using models such as neural networks in which one doesn't know the process of its decision-making, can make it hard to understand and tricky to trust its predictions. Lastly, consideration of how these predictions are being used is vital. A model restricted to only big investors can make it hard for others to compete. To keep things fair, it is important to use data encompassing a wide range of companies and situations while regularly checking on the models to confirm that they aren't showing any hidden biases.

Aside from biases in data, market volatility is also an important factor. Volatility is the degree of variation in stock prices over time due to external factors such as economic changes, geopolitical events, or investor sentiment. It plays an important role in predictions for stocks. The models in this study indirectly include this factor by using training data that captured patterns in price fluctuations due to volatility over time. However, by studying the performances of the models, some models handle volatility better than others. For example, Tesla and Nvidia, known for high volatility, exhibited higher Mean Squared Error (MSE) values across all models compared to more stable stocks like Costco and Nike. When analyzing models, Neural networks performed best during testing, highly likely due to their ability to recognize complex patterns. In comparison, Decision Trees and Random Forests overfit the training data, making them less adaptable to sudden changes. Additionally, the trading bot's decision-making process adjusts stock purchases based on next-day price predictions, which acts as a simple risk management tool in volatile conditions. The bot sells stocks when price drops are predicted, helping minimize potential losses due to unstable market conditions. Furthermore, the study also uses cross-validation to ensure that models are not too dependent on past trends, helping to generalize to new unseen market conditions. While models do not explicitly state volatility indicators, their varying performances and influences suggest an indirect use of such indicators.

Interpretability is also a critical factor in stock prediction, especially when it concerns financial decision-making. Understanding how and why a prediction is made is important as it can affect your decisions in investments. Out of all the models used in the study, Linear Regression and Decision Trees

are the most interpretable. Linear Regression provides clear coefficients that show how each independent variable, in this case, past stock prices, influences the predicted prices, making it easy to understand. Decision Trees use a step-by-step branching structure where each decision is based on simple conditions, allowing for transparency in how predictions are made. However, Neural Networks and Random Forests are less transparent. Neural Networks operate using multiple hidden layers and weight adjustments, making it difficult to trace how the input data produced the final prediction. This "black-box" nature raises concerns in financial markets where transparency is necessary for risk assessment. Random Forests, while reducing overfitting by combining multiple Decision Trees, become less interpretable as they combine the outputs of numerous trees, making it a difficult model to understand which factors drove the prediction. For financial decision-making, interpretability plays a crucial role. Investors and financial analysts often prefer models that can provide justifiable predictions over highly complex models with superior accuracy but little transparency. This trade-off between accuracy and explainability is important when selecting models for real-world investment strategies.

The trading bot developed in this study follows a simple investment strategy: stocks are purchased when the model predicts a price increase and sold when a decrease is predicted. To evaluate the effectiveness of this strategy, backtesting was conducted using historical stock data from the past five years. The model-generated predictions were compared to actual stock movements to determine whether the bot's trades would have resulted in profits or losses. Several assumptions were made in this simulation, including Transaction Costs, Liquidity, and Market Timing. The study assumes no brokerage fees or taxes on stock trades. In real-life applications, incorporating transaction costs impacts the net profitability of the strategy. The model also assumes that stocks can always be bought or sold at the predicted prices without significant market impact. This may not be true for large-volume trades or illiquid stocks. The bot executes trades based on the next day's predicted opening price, assuming immediate execution at that price. However, real-world fluctuations between market openings and closings may introduce slippage, affecting trade outcomes. While the backtesting results suggest that investing based on model prediction can lead to financial gains, these assumptions limit the impact of the performance in the real world. Future implementations could refine the strategy by incorporating strategies to counter the limitations mentioned above by including transaction costs or alternative trading signals.

## Methods

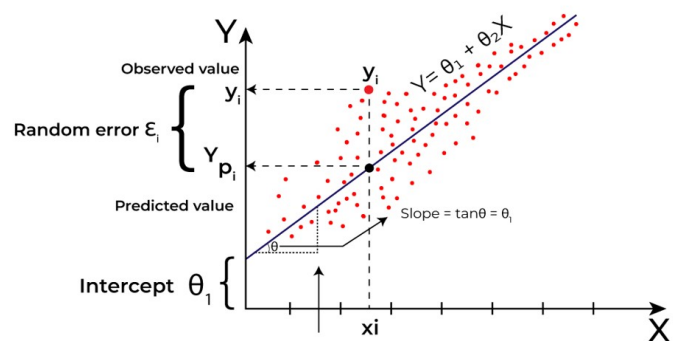
To be able to make a model that predicts stock prices by taking the input of the last three days' open prices, there needs to be data to use. In this case, Yahoo Finance is used to get data for

daily opening stock prices for the last 1260 days, or 5 years, when the stock market was open. While data was collected, processing was required before it could be implemented for training. Luckily, there weren't any encounters with any missing data so this step in the processing was skipped. Next, any extra data, that isn't required is eliminated. Only the 'Open' price column is retained and the rest of the features are set aside. After eliminating the excess data, the X and Y variables for the code are defined. A set of input and output values is created where the stock price for the three days before the predicted stock price is the X value, and the price on the fourth day (the predicted price) is the Y value.

For this research, daily opening stock prices over five years (1260 trading days) were chosen as the dataset. The decision to use daily data was made to capture short-term stock price movements and allow for a more responsive trading strategy. Using daily prices instead of weekly or monthly data provides more frequent updates for predictions, which is valuable for active traders. However, the choice of daily data comes with certain limitations. Since only opening prices were used, intra-day price movements, closing prices, and after-hours trading fluctuations were not accounted for. This could lead to missed opportunities in price patterns that develop throughout the trading day. Additionally, while daily data captures more granular trends, it can introduce noise, making it harder to distinguish meaningful patterns from short-term fluctuations. The impact of different timeframes on stock price predictions is another consideration. Short-term models that use daily data may be highly responsive yet more sensitive to market volatility, whereas longer-term models using weekly or monthly data could smooth out fluctuations but may fail to capture rapid price changes. Future research could explore alternative timeframes, such as weekly or monthly stock data, to assess whether longer-term trends provide improved predictive accuracy. Also incorporating high, low, and closing prices instead of just opening prices could enhance the model's ability to detect more complex price movements.

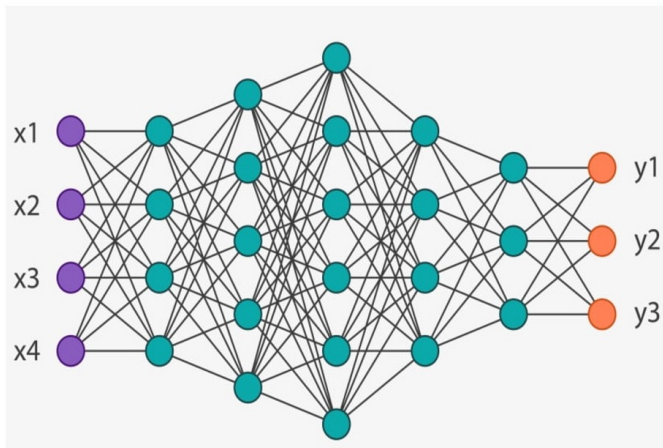
Now, it is time to split the data into three groups: the training set (70% of data), the validation set (15% of data), and the testing set (15% of data). The purpose of this is to prevent the model from overfitting data (when a model is trained to fit training data too closely, causing it to be unable to make accurate predictions on new data) and to be able to evaluate the model's actual performance when faced with new data that it has never seen before. The training data is used to help the model learn the patterns and relationships between the data, allowing it to be able to make predictions. The validation set is used to evaluate and fine-tune the model so it can be hyperparameter-tuned to improve its performance. Hyperparameter tuning is the process where different hyperparameter values are tested to find the best ones that optimize the results from the model. The testing data is only used once and that is to test the performance of the model when it has never been seen before.

Linear regression is a statistical method that identifies relationships between independent and dependent variables. It can be used in machine learning to predict the value of a dependent variable (y value or output) when given one or more independent variables (x value or input). Linear regressions are suitable for modeling linear trends in stock prices due to their sustainability and simplicity. The model requires minimal computational needs, making it easier to understand and use. To find a prediction, the method tries to find a straight line that best fits with the data points it is given. There are two types of linear regressions, Simple Linear Regression and Multiple Linear Regression. Simple Linear Regression where there is only one independent and one dependent variable. Multiple Linear Regression is when there are multiple independent variables and one dependent variable. This is also the one used for the research. The multiple independent variables were the past three days of stock prices and the dependent variable was the predicted stock price for the next day. When used for predictions, the model showed reasonable performance for general trends but struggled with highly volatile data.

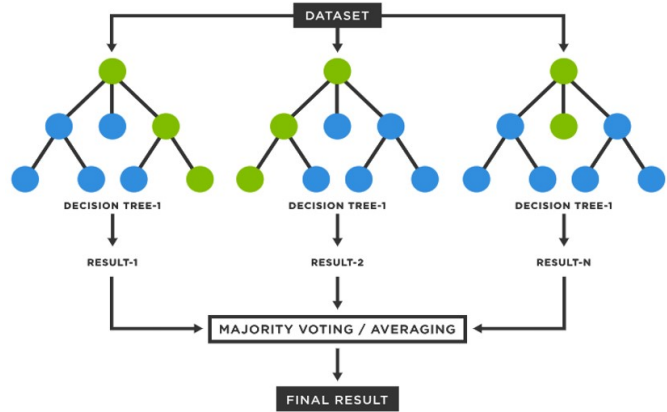


**Fig. 2** Linear regression showing the predicted line of best fit, with observed data points and random

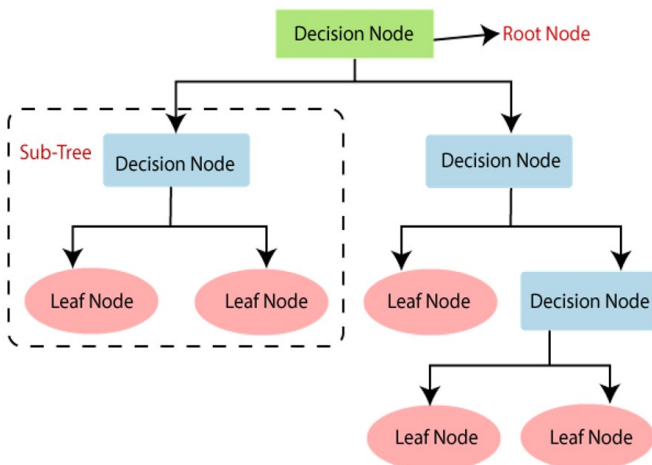
A neural network is a type of machine-learning model that is inspired by the human brain in the sense that it processes information in layers to identify non-linear patterns in stock data and learn from the data. The layers are comprised of interconnected nodes (neurons) where each node receives input signals from the layer before it. Then, the node performs a calculation on the input and passes it onto the next layer until it reaches the end and one can see the results. This allows computers to perform complex jobs such as Image Recognition, Natural Language Processing (NLP), and even Speech Recognition. The neural network is particularly useful and has great potential for recognizing hidden trends in volatile stock markets, as well as analyzing data that can be valuable in making predictions. However, it is computationally expensive due to multiple layers and iterative backpropagation, making it time-intensive. Using a complex model in our simulation, helped text various levels of predictive power in which neural networks performed



**Fig. 3** Neural network with multiple layers



**Fig. 5** A representation of an ensemble of decision



**Fig. 4** Decision tree

well in capturing non-linear relationships but showed signs of overfitting on historical data.

A decision tree is a machine-learning algorithm that has a tree-like structure, hence the name. It makes predictions by asking a series of questions and branching off based on the answers until the "leaf" node is reached, also known as the final decision or output. It is used for both classification purposes and regression purposes since it is considered a powerful tool due to its interpretation and visualization abilities. However, this model is susceptible to overfitting and has a moderate computational load which grows with tree depth. For this specific research, this model is used for regression purposes to predict future stock prices. It performed exceptionally in training, but it is evident from its near-zero mean squared error for certain companies, like Tesla, that it was overfitted.

The Random Forest is a machine-learning algorithm that shares many similarities with the decision tree. It operates as a "forest" of decision trees and is made up of trees that are trained

using a random subset of data. Random forests have a higher complexity due to training multiple trees and combining outputs but are justifiable for financial prediction tasks. The combined output of these trees used to make more accurate predictions reduce the overfitting factor of decision trees. This technique of combining the "best" of individual trees is called ensemble learning and is effective for both classification and regression. For our research, it is being used for regression purposes as it is needed to predict future stocks since it can generalize better for volatile financial datasets. In the research, it was concluded that Random Forests balanced the robustness during testing, making it more reliable for stock market unpredictability.

The models described above were picked carefully with thought and consideration for the purpose. Linear regression was chosen for its simplicity and efficiency in identifying linear relationships, hence its name. It is a well-established baseline model for regression tasks, including financial time series. Neural networks, on the other hand, are selected for their ability to model non-linear and complex patterns in data, especially useful for stock price fluctuations. It has been used in similar studies for predicting financial trends due to its adaptability. Decision trees have great interpretability and simplicity in identifying key decision points in data and are effective in regression tasks that may not be strictly linear. Random Forests are chosen for their ensemble nature, which reduces overfitting by aggregating predictions from multiple decision trees. It is known for its robustness and high performance in financial prediction studies. All in all, different models represent a mix of simple (Linear Regression) and complex (Neural Networks) approaches to test various levels of predictive power. Decision Trees and Random Forests are known for interpretability and ensemble accuracy, which is relevant for financial datasets.

The next step is to use the X\_train and Y\_train as variables and train the models (linear regression, neural network, decision tree, and random forest). While training the models, hyperparameter

---

tuning is also performed to maximize the performance of certain variables in each model. A few metrics are used to calculate the models' performance.

The first one to be addressed is the mean squared error. This evaluates the average squared difference between predicted and actual values, essentially the quality of a model's predictions. Lower values indicate better model performance. This metric is crucial as it penalizes larger errors more heavily, which is essential in financial modeling as it affects each model fairly according to the results it gives.

The second metric used is Mean Absolute Percentage Error or MAPE for short. MAPE measures the average percentage error between predictions and actual values, which is useful for understanding relative prediction accuracy. Since it is a percentage-based error, it is easier for people to understand and compare.

After training the models, it is necessary to figure out how to combine the best models to achieve the best prediction. To do this, a method called Multiplicative weight update is applied. It is where the "importance" or "weight" of different options (like predictions from different models) is adjusted based on how well they performed in the previous round.

The multiplicative weight update uses the prediction error for each of the models to update their weight in the final prediction. Initially, all models start with equal weight in the results. Using the test data points, the algorithm calculates the error between the prediction and the actual values to adjust the weight of a model by  $(2^{\text{error}})$ . After adjusting the weights, the values are normalized, by dividing each by the sum of all weights, to ensure that they still add up to one. By using this method, the models that make more errors are penalized, but they are not eliminated when making the final decision.

After developing the models, a trading bot is programmed to simulate the potential profit from investing a specific amount of money in a certain number of stocks of a company, using the models to predict the next day's stock performance. The trading bot prompts the user for the investment amount and offers various options for the number of stocks to be bought or sold over 5 years. The bot has specific conditions for buying and selling stocks. It uses a linear model to predict the next day's opening stock price and uses this information to make decisions regarding the stocks. For example, if the predicted stock value for the next day is higher than the current value and the user has enough funds, the bot will purchase the stock to hopefully be able to gain profit from the stock's increased value. The code automatically sells stocks if there is a prediction of potential loss and buys them if there is a prediction of profit for the next day.

Upon completion of all necessary tasks, a program designed to predict the following day's stock price will be created. This program utilizes all accumulated codes and resources, requiring the input of a company's stock prices for the preceding three

days to predict the next day's price with reasonable accuracy.

While historical stock prices provide a strong foundation for understanding and utilizing past trends, stock prices are also influenced by numerous other factors such as economic policies, geopolitical events, and news sentiment. These events cannot be captured by the use of historical data alone. Integrating the data with current news and other economic indicators will further enhance the accuracy and reliability of these predictions.

Investing in financial markets always involves some risk. Sometimes, the risk is low; other times, it is high, and investing is a gamble. Using predictive models to help with your investment choices also has risks that must be managed. The models used in this study focus on historical stock prices which provide valuable insight but are not immune to day-to-day changes of the uncertain market conditions. To lessen this uncertainty, additional risk management techniques need to be enforced. A few examples of this include metrics such as the Sharpe ratio and maximum drawdown. The Sharpe ratio measures risk-adjusted returns and adjusts the performance for the excess risk the investor takes. The maximum drawdown measures an asset's largest price drop from a peak to a trough. It is also considered an indicator of downside risk. By adding these techniques, a clearer picture of model performance in real-world scenarios is provided. However, cross-validation was employed in this study, which reduced overfitting, ensured strong predictions across multiple datasets, and minimized prediction errors.

Additionally, the trading bot that is a part of this research demonstrates an application of risk mitigation, which is the process of actively reducing potential negative impacts of identified risks by developing strategies to manage and limit setbacks. Adding the feature of dynamically adjusting stock purchases and sales according to predicted prices minimizes potential losses. Future iterations of this work could incorporate strategies such as stop-loss orders, portfolio diversification, and sentiment analysis which can further enhance returns by adjusting the risks accordingly. Another technique to implement in the future would be stress-testing the models under extreme market conditions such as sudden crashes or economic shifts. This would provide valuable intel into a model's applicability to make it practical for real-world investments.

This research explored the use of machine learning models in predicting stock prices and compared different approaches to determine their effectiveness. While the models showed promising results, there are still areas for improvement that future research can address. One important next step is to expand the dataset by incorporating additional sources of information, such as financial news, social media sentiment, and macroeconomic indicators. These factors influence stock prices in real-time and could help improve the accuracy of predictions.

Another area for improvement is testing more advanced machine learning models, such as Long Short-Term Memory (LSTM) networks or Transformer-based models, which are de-

---

signed to recognize patterns in time-series data. Additionally, implementing standard financial risk metrics, such as the Sharpe ratio and Sortino ratio, would help measure how well the models balance potential returns with risk.

Future research could also focus on applying these models to different stock markets, such as European or Asian markets, to see if they generalize well across different economic conditions. Testing model performance across different periods, including major financial events like recessions, would also help determine how adaptable they are.

Finally, improvements can be made to the trading bot by incorporating real-world investment factors like transaction costs, liquidity constraints, and market timing issues. These additions would make the bot's performance more reflective of real trading conditions. By addressing these areas, future research can improve stock prediction models, making them more reliable and practical for investors looking to make informed financial decisions.

### Acknowledgments

I would like to thank my father Mr. Amit Varshney for his constant support and motivation throughout my research.

### References

- 1 R. Jain and R. Vanjara, *Emerging Trends in AI-Based Stock Market Prediction*.
- 2 R. Chopra and G. Sharma, *Application of Artificial Intelligence in Stock Market Forecasting J*, <https://doi.org/10.3390/jrfm14110526>.
- 3 H. Htun, M. Biehl and N. Petkov, *Forecasting relative returns for SP 500 stocks using machine learning*, <https://doi.org/10.1186/s40854-024-00644-0>.
- 4 N. Rauf, M. Malik, T. Arif, S. Sharma, S. Singh, S. Aich and H. Kim, *Stock Market Prediction Using Machine Learning Techniques: A Decade Survey on Methodologies, Recent Developments, and Future Directions*.