International Journal of Science Academic Research
Vol. 05, Issue 04, pp.7357-7364, April, 2024
Available online at http://www.scienceijsar.com

ISSN: 2582-6425

Research Article

STOCK MODEL PREDICTION RESEARCH

***Navye Vedant**

United States

Received 17th February 2024; **Accepted** 24th March 2024; **Published online** 30th April 2024

Abstract

Stock investment prices are never still; they are always changing. It is important to stay informed on the upward or downward trends of the market to make future investments. This paper aims to examine the question: which of the python models used in this study are the most accurate at predicting the price of the stock market, *x* days into the future? To accustom the machine learning (ML) predictor to the multitude of possibilities that could categorize stock patterns, 7 different ML models were trained on 1250 pieces of open stock market data dating to the last 5 years by assigning weight values to all the models based on their accuracy. Results showed that two of the ML models, specifically the Linear Regression and the Random Sample Consensus (RANSAC) Regress or models consistently outperformed the other 5 models, both ending up with the highest weight values of around 0.5 when predicting for Amazon, Apple, and Tesla. Therefore, the RANSAC and Linear Regression models are the best models to rely on when predicting open stock market prices using ML.

Keywords: Machine Learning, Artificial Intelligence, Random Sample Consensus, Line of Best Fit, Support Vector Regression, Mean Squared Error.

INTRODUCTION

This project focuses on crafting a machine learning (ML) stock prediction model for three different companies' (Apple, Amazon, and Tesla) stock using Python on Google Collab. The dataset is a collection of numerical data, specifically historical open stock prices for each company. This paper uses 1250 open stock price samples each for Amazon, Tesla, and Apple, every piece of data representing a different point date (Macrotrends, 2024) . To see how well the model performs, the dataset was carefully split. Around 33% (417 samples) are set aside for the testing dataset, while the remaining 67% (833 samples) are used for training. The dataset is entirely made up of historical open stock prices from the last 5 years. These are numbers necessary to predict what each stock will be valued in the future. Open and close prices give a peek into daily stock performance, while high and low prices show how prices change throughout the day (Investopedia, 2023). The trading volumes show how much activity and interest there is in a company's stock. This research holds paramount importance in advancing our utilization of artificial intelligence to predict economic factors, notably within the dynamic domain of the stock market. As AI technology evolves, the potential for enhanced stock trend forecasting becomes increasingly significant. The primary objectives focus on determining the optimal performance among the seven machine learning models employed. However, the study acknowledges inherent limitations, particularly in achieving uniform performance across companies due to distinct stock trend shapes. Despite these challenges, the research contributes valuable insights to the integration of AI in finance, emphasizing both its potential and the strategic considerations necessary to navigate hurdles. To make accurate predictions, it's important to use a sophisticated approach. Seven different ML models were trained, including linear regression, RANSAC, SVR, Gaussian regression, Random Forest Regressor, a neural network, and Decision Tree model. Each model was given a "performance score", or a weight, based on how well they've done historically, using an approach that combines multiple models for better results. The model with the best track record holds the most credibility to provide predictions on stock prices. One alternate method that has been historically employed to predict the market is the "Elliot Wave theory" (Investopedia, 2023) . This theory is a form of technical analysis that attempts to predict future price movements by identifying patterns in market sentiment. It is based on the idea that financial markets move in cycles and that these cycles can be analyzed and predicted. The Elliott Wave Theory suggests that markets move in waves, with alternating patterns of upward and downward movement. These waves are subdivided into impulsive waves (trending upward) and corrective waves (retracing the trend). Traders use this theory to identify the current wave and predict the next one. The difference between the Wave Theory and the stock prediction model is that market sentiment isn't taken into consideration when using the stock prediction model. Other papers such as the one done by (Wong, Figini, Raheem, Hains, Khmelevsky, & Chu, 2023) used market sentiment calculations into their predictive processes. As the paper suggests, this can be a good and a bad thing:

Using sentiment

Pros:

Providing a structured framework for understanding certain trends. Can help traders anticipate potential turning points in the market.

Cons:

Highly subjective and open to interpretation, making it challenging to apply consistently. Not always accurate, as market sentiment can be influenced by various factors. The paper acknowledges that their market sentiment analysis can

Corresponding Author:** ***Navye Vedant
United States.

be better, but they lacked the resources to do so currently. For the best possible results, this paper does not make use of market sentiment analysis, to provide as standardized and reliable of information as possible.

This paper is organized as follows:

1) Introduction: This section offers a concise overview of the research paper's scope and objectives.
2) Literature Review: Identifying the gaps in research about AI technology and its advancement into market calculations and statistics.
3) Models and Methodology: This segment displays the comprehensive analysis of seven distinct models utilized in this study. It details their functionalities and potential contributions to the research.
4) Results: A presentation of comprehensive data tables that elucidates the efficacy of all seven models in predicting open market prices based on the provided training data. The analysis prioritizes understanding high-performing models.
5) Conclusion: This concluding section delves into the implications of the findings on the future landscape of stock prediction and machine learning models. It also explores potential avenues for further research.

LITERATURE REVIEW

The literature review section aims to contextualize the evolution of artificial intelligence (AI) and machine learning (ML) in stock price prediction research, highlighting the existing gaps in predicting unpredictable market behaviors. It endeavors to showcase how this research paper seeks to fill this gap by employing AI-infused ML models to forecast aspects traditionally deemed unforeseeable in financial markets.

Context

The rapid advancements in artificial intelligence (AI) and machine learning (ML) have revolutionized various industries. AI, particularly ML, has shown promise in predictive analytics, demonstrating the capability to forecast complex outcomes. Recent years have witnessed a surge in AI applications, leveraging vast datasets and sophisticated algorithms to make predictions in domains ranging from healthcare to finance.

AI and ML in Stock Price Prediction

Studies utilizing AI and ML for stock price prediction have proliferated, aiming to leverage these technologies' potential in forecasting financial markets. Existing literature demonstrates a diverse range of ML models applied to predict stock prices, including neural networks, regression algorithms, decision trees, and alternate methods. However, despite these efforts, the financial markets' inherent complexity and unpredictability persist as challenges in achieving consistent accuracy in stock price forecasting.

Addressing the Gap in Literature

The prevailing literature acknowledges the limitations of traditional models in forecasting stock prices accurately, especially in capturing unpredictable market behaviors. There exists a critical gap where the application of AI and ML to predict inherently unpredictable events or trends—such as sudden market shifts or anomalies—is limited. Current research tends to focus on historical data analysis and pattern recognition, often falling short in handling unforeseen market dynamics. This research paper aims to bridge this gap by employing existing ML models within the realm of AI to predict aspects of the stock market traditionally considered unpredictable. Leveraging historical data and advanced ML techniques, this study endeavors to explore the application of diverse ML algorithms to foresee trends that were conventionally deemed impervious to prediction. By amalgamating AI capabilities with established ML methodologies, this research strives to break new ground in forecasting market behaviors previously deemed uncertain or unforecastable. By leveraging the advancements in AI and expanding the scope of traditional ML models, this research seeks to pioneer a more comprehensive approach to stock price prediction. Through this exploration, the goal is not just to enhance predictive accuracy but also to shed light on the potential of AI in forecasting elements previously considered beyond prediction.

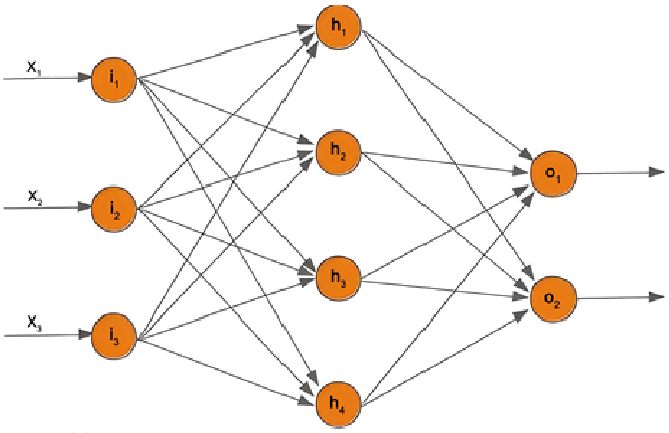
MODELS AND METHODOLOGY

This research project made use of 7 different models, consisting of Neural Network, RANSAC Regressor, Decision Tree, Random Forest Regressor, Gaussian regression, Linear Regression and Support Vector Regression models in order to predict stock market prices. Each model's individual functions are explained below.

Neural Network

This model leverages the power of artificial neural networks, which are designed to mimic the human brain's interconnected neurons, to make sense of complex data and make informed predictions. The neural network demonstrates several strengths that contribute to its effectiveness in predicting stock prices. One of its significant advantages lies in its ability to capture intricate, non-linear relationships within the data. This allows it to grasp subtle patterns and dependencies that may elude traditional linear models. Additionally, neural networks are highly adaptable and can learn from a vast amount of historical data, making them particularly well-suited for stock market predictions where historical patterns play a crucial role (Hardesty, 2017). However, it is essential to recognize that neural networks also have their weaknesses. One notable limitation is their "black-box" nature (Colah, 2014), which can make it challenging to interpret their decision-making processes. It may be difficult to discern exactly why the model arrives at a particular prediction, which could be a drawback when transparency and accountability are critical. Neural networks also require a substantial amount of data and computation, and they can be sensitive to over fitting if not properly regularized and validated. To dive into the specifics of how the neural network model operates, it employs a multi-layered architecture. Each layer consists of artificial neurons that process and transform the input data. These neurons are interconnected with weighted connections that hold the key to the model's predictive power (see Figure 1). The neural network assigns weight values to these connections during a training phase, where it learns from historical stock price data. This process involves minimizing a loss function, such as mean squared error, which quantifies the model's prediction accuracy. The model iteratively adjusts these weights using

optimization algorithms like gradient descent. Furthermore, the neural network employs activation functions within each neuron, which introduce non-linearity into the model. This non-linearity allows the network to capture complex relationships within the data. By passing the weighted sum of inputs through these activation functions, the neural network can learn intricate patterns, making it a valuable tool in stock price prediction. To determine the significance of data points, the neural network considers the magnitude of the weighted connections. Weight values represent the model's belief in the importance of specific input features. Data points with higher-weighted connections are considered more influential in making predictions, while those with lower weights have less impact.

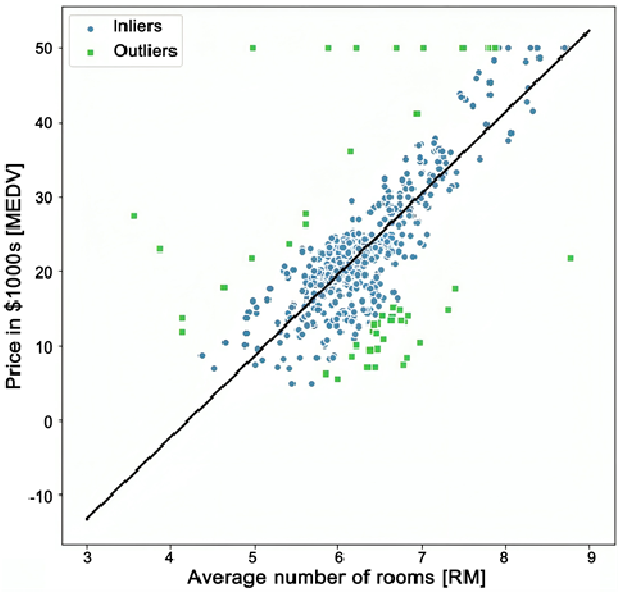


Source: Klein, 2023

Figure 1. Depiction of how a neural network operates within a weightage system, "i" representing input layer and "h" representing the next layer known as "hidden layer"

RANSAC Regressor

This ML model employs the Random Sample Consensus (RANSAC) algorithm, primarily designed for robust regression analysis. The RANSAC regressor offers a set of unique strengths that contribute to its effectiveness in stock price prediction. One of its standout features is its robustness against outliers. It's well suited for situations where the data may contain noise or erroneous data points that could mislead traditional regression models. RANSAC identifies and disregards these outliers, ensuring that the model's predictions are not unduly influenced by them (see Figure 2). Furthermore, the RANSAC regressor is adaptive and versatile, making it capable of handling different data distributions. It excels in cases where linear regression models might fail due to non-linear relationships in the data. This adaptability ensures that it can be applied to a wide range of stock data scenarios (Kumar, 2020). However, it's essential to recognize that the RANSAC regressor also has its limitations. While it's robust against outliers, it may struggle when the proportion of outliers is exceptionally high. In such cases, it could fail to find a meaningful consensus and may not produce reliable predictions. Additionally, the algorithm's performance can be affected by the choice of parameters, such as the maximum distance for an inlier or the number of iterations. Now, delving into the specifics of how the RANSAC regressor operates, this model works by iteratively fitting a regression model to a subset of data points, known as a random sample. The consensus set is formed by identifying data points that are close enough to the regression line, given a predefined threshold.

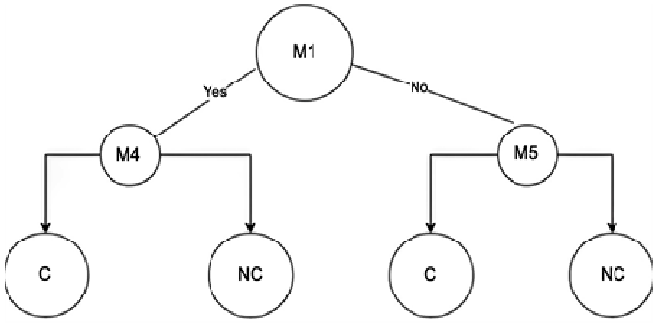


Source: Kumar, 2020

Figure 2. RANSAC regressor line of best fit (LOBF)

Decision Tree

Decision tree utilizes a tree-like structure to make decisions, and it has proven to be a versatile tool in predicting stock prices (see Figure 3). The Decision Tree model boasts several strengths that contribute to its effectiveness in stock price prediction. One of its primary advantages lies in its interpretability. The Decision Tree is designed to provide a clear and understandable decision-making process. This transparency is invaluable, as it allows users to grasp why the model arrived at a particular prediction, making it a practical tool for financial analysis. Additionally, Decision Trees can handle both numerical and categorical data, making them adaptable to a wide range of financial datasets. They are robust to outliers and do not require extensive data preprocessing, simplifying the modeling process. This adaptability ensures that Decision Trees can be applied effectively in the stock prediction context. However, Decision Trees do have some limitations. One significant drawback is their tendency to overfit the training data, which can result in poor generalization to new, unseen data. To mitigate this, model pruning and other techniques are often employed to improve performance. Another challenge with Decision Trees is their inability to capture complex, non-linear relationships in the data, which can be a limitation in scenarios where such relationships are critical. In this experiment, model pruning and fit adjusting techniques were not used.



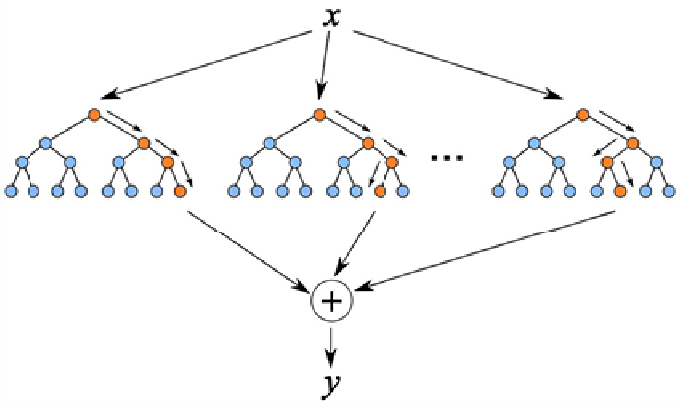
Source: Author, 2023

Figure 3. Decision tree model

To delve into the specifics of how the Decision Tree model operates, it is constructed as a treelike structure with nodes and branches. Each node represents a decision point based on a feature or attribute, and each branch leads to a further decision node or a final prediction.

Random Forest Regressor

This ML model is a powerful tool known for its ability to handle complex data and provide accurate predictions in a diverse range of scenarios. The Random Forest Regressor boasts several strengths that contribute to its effectiveness in stock price prediction. One of its primary advantages is its capacity to capture complex relationships in the data. This is achieved through a collection of decision trees, which collectively work together to make predictions (see Figure 4). Each decision tree is trained on a subset of the data, allowing them to capture different aspects of the dataset's complexity. By averaging the predictions of these trees, the Random Forest model achieves robustness and adaptability, which are essential qualities in the stock prediction domain. Another notable strength is Random Forest's ability to handle both numerical and categorical data without the need for extensive data preprocessing. This makes it a versatile choice when dealing with financial datasets that often consist of various data types. However, it's essential to recognize that the Random Forest Regressor also has its limitations. While it excels at capturing complex relationships, it might not always provide the same level of interpretability as a single Decision Tree model. The collective decision-making process of multiple trees can make it challenging to discern the exact factors driving predictions.



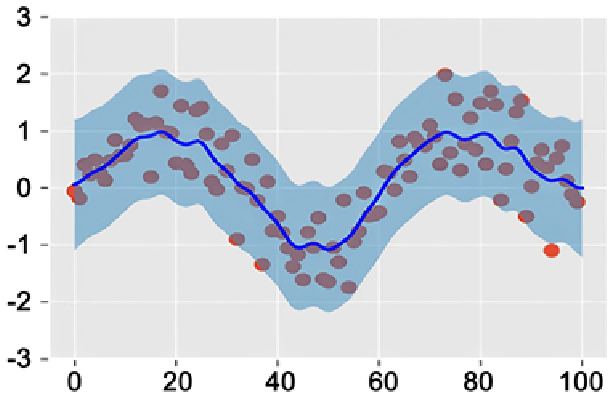
Source: Mwiti, 2023

Figure 4. Random forest regressor model where "x" represents input value

Additionally, like other methods, the Random Forest is not immune to overfitting, although it is generally more resilient due to the combination of trees. To provide an in-depth explanation of how the Random Forest Regressor operates, this model utilizes a plethora of decision trees. Each tree is constructed by selecting a random subset of data and a random subset of features. The randomness introduced in building these trees helps to mitigate overfitting and promotes diversity (Sahai, 2023). Random Forest assigns weight values to the individual decision trees based on their performance. Trees that contribute more accurate predictions are assigned higher weights, signifying their significance in the model's decision-making process. Weights are calculated by evaluating the mean squared error or another relevant performance metric for each tree.

Gaussian Regression

The Gaussian Regression model boasts several strengths that contribute to its effectiveness in stock price prediction. One of its primary advantages is its ability to capture and model data distributions effectively. By assuming that the data follows a Gaussian distribution, the model can make predictions while accounting for the inherent uncertainty in financial markets. This makes it particularly well-suited for situations where stock prices exhibit relatively normal, bell-shaped distribution patterns (see Figure 5). Additionally, the Gaussian Regression model provides a natural framework for probabilistic predictions. It not only offers point estimates for stock prices but also provides confidence intervals, which can be crucial for risk assessment and portfolio management. This strength enhances its applicability in financial analysis and decision-making. However, it's important to acknowledge that the Gaussian Regression model also has its limitations. It assumes that the data follows a Gaussian distribution, which may not always hold true in real-world scenarios. In situations where stock prices exhibit non-Gaussian behaviors, the model's accuracy can be compromised (Xu, Kuplici, Sen, & Paulus, 2021) . This limitation requires careful consideration when applying the model to different types of financial data.



Source: Sander, 2021

Figure 5. Gaussian regression curve

To delve into the specifics of how the Gaussian Regression model operates, it employs the principles of linear regression but adds a probabilistic twist. Instead of providing a single point estimate for stock prices, it generates a probability distribution of potential outcomes. This distribution is typically Gaussian, with a mean and variance that describe the central tendency and the level of uncertainty in the prediction.

Linear Regression

Linear Regression boasts several strengths that contribute to its effectiveness in stock price prediction. One of its primary advantages is its simplicity and interpretability. The model operates on the assumption that the relationship between input features and stock prices is linear, making it intuitive to understand. This transparency allows us to easily interpret the coefficients associated with each feature, which provides insights into their influence on the stock price. Additionally, Linear Regression is computationally efficient and quick to train, making it a practical choice when dealing with large datasets. It's also a robust choice for modeling situations where the relationship between features and stock prices is, indeed, linear, as is often the case in finance. Unfortunately, it is inherently limited in capturing complex, non-linear

relationships within the data. In scenarios where stock prices exhibit non-linear patterns, the model's predictive accuracy can be compromised. This limitation calls for careful consideration when applying Linear Regression to financial datasets, as it may not be the ideal choice for all scenarios.

To provide an in-depth explanation of how Linear Regression operates, the model establishes a linear equation relating the input features to the stock price. This Equation (1) takes the form of:

$$Y=\beta0+\beta1X1+\beta2X2+\cdots+\beta nXnY=\beta0+\beta1X1+\beta2X2+\cdots+\beta(1)$$

where:

Y represents the predicted stock price.

$X_1, X_2, ..., X_n$ are the input features.

$\beta_1, \beta_2, ..., \beta_n$ are the coefficients associated with each feature.

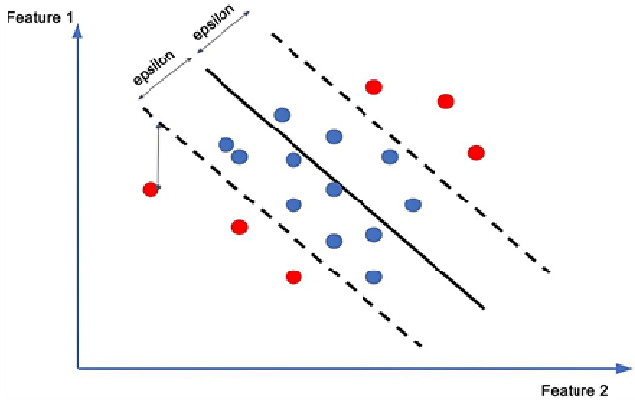
To assign weight values to the Linear Regression model, the coefficients $\beta_0, \beta_1, \beta_2, ..., \beta_n$ are evaluated. Coefficients that hold higher absolute values are indicative of features that have a more significant impact on the stock price. These coefficients effectively serve as the weights assigned to the respective features, signifying their significance in the model's decision-making process.

Support Vector Regression

The Support Vector Regression (SVR) model boasts several strengths that contribute to its effectiveness. One of its primary advantages is its capacity to handle both linear and non-linear relationships (Singh, 2023) between input features and stock prices. It achieves this by mapping the input data into a higher dimensional space and finding the optimal hyperplane that best fits the data. This flexibility is crucial in financial markets, where relationships between variables can be intricate and dynamic. Another notable strength of SVR is its robustness to outliers. It focuses on finding the best-fitting hyperplane while allowing for some degree of error, which helps mitigate the influence of extreme data points that might distort predictions. This robustness enhances the model's reliability in situations where outliers are prevalent in financial data. One limitation lies in the choice of kernel functions used for mapping data into a higher dimensional space. The model's performance is highly dependent on the selection of an appropriate kernel, which can be a challenge in practice. Additionally, SVR can be computationally intensive, particularly when dealing with large datasets, which can impact its efficiency. To provide an in-depth explanation of how Support Vector Regression operates, the model seeks to find the optimal hyperplane that minimizes the margin of error for stock price predictions. This margin is defined by a parameter known as "ϵ" (epsilon) (see Figure 6), and the goal is to maximize the margin while allowing for some deviations from the hyperplane. The model also employs kernel functions to transform the input data into a higher-dimensional space, where it becomes more amenable to linear separation.

METHODOLOGY

To effectively single out the most successful model out of these 7, the study uses a weightage calculator, coded on python. The code initializes weight values for the seven different models to a value of 1. It then enters a loop that iterates through the elements of a dataset, the training data, "X_test". Within this loop, the code calculates the absolute errors for each model's prediction compared to the actual open market price values in the "Y_test" dataset and adds 1 to these errors. These adjusted errors are then used to update the weights for each model. The weights are updated inversely proportional to the absolute error, meaning that models with lower absolute errors will receive higher weights, and vice versa. After updating the weights for each model, the code calculates the sum of all weights. Then, it normalizes the weights by dividing each weight by the sum. This normalization ensures that the sum of all weights is equal to 1, making them a valid set of proportions or percentages.



Source: Author, 2023.

Figure 6. SVR graph with two epsilon parameters

Finally, the code prints out the normalized weights for each model along with a label identifying the model. These weights represent the relative importance of each model in prediction, with higher weights indicating more trust in the predictions made by that model.

RESULTS AND DISCUSSION

These models were assessed using a range of metrics and hyperparameters to gauge their performance and reliability in forecasting stock prices. They were tested against the metrics of three different companies: Amazon, Apple, and Tesla. All of the integers used in these data tables are real data which corresponds to the month of October 2023. The mean squared error, or MSE, displays the squared difference between the actual value and the predicted value. Squaring the differences ensures that negative and positive errors do not cancel each other out. It emphasizes larger errors and penalizes them more significantly. The weight is determined solely through the testing data, which is why the MSE training data will not affect the weightage at all. The weightage is therefore determined based on which model had the least mean squared error for the testing dataset. In simple terms, the MSE being higher is a indicator of negative performance, and the weight being higher in value is an indicator of positive performance. See Table 1 for a sample table. The table above displays some fake data for the sake of an example. Analyzing this table, the MSEs are all relatively low except for Amazon, which causes the average to rise significantly. Apple and Tesla's MSE are all relatively low, meaning that they have made relatively little mistakes in their testing and training data. This model performed at a subpar level at predicting Amazon's stock, but is still reliable when predicting for the others. The average weight of this specific fake model was 0.147, meaning that out of all of the 7 models, this specific model carried 14.7% of the weight. Testing refers to the testing data (417 samples) and training

refers to the training data (833 samples). All of these samples are real open stock market prices from the last 5 years, which are used to attempt to predict the newest price with the least error below are the results of all 7 tables. Table 2 corresponds to the Neural Network model, Table 3 with the RANSAC regressor,

Table 1. Fake data - sample table

COMPANY	MEAN SQUARED ERROR (Testing)	MEAN SQUARED ERROR (Training)	WEIGHT
TESLA	2.3	2.08	0.44
APPLE	4.5	22.3	0.002
AMAZON	77.8	65.7	0.00003
AVERAGE	28.2	30.027	0.147

Table 2. Neural network

COMPANY	MEAN SQUARED ERROR (Testing)	MEAN SQUARED ERROR (Training)	WEIGHT
TESLA	73.97	61.31	0.00013
APPLE	5.98	8.04	$2.99e^{-11}$
AMAZON	9.84	11.21	$9.79e^{-13}$
AVERAGE	29.93	26.85	$4.333e^{-5}$

Table 3. RANSAC regressor

COMPANY	MEAN SQUARED ERROR (Testing)	MEAN SQUARED ERROR (Training)	WEIGHT
TESLA	71.50	59.17	0.49993
APPLE	4.80	6.30	0.5
AMAZON	7.85	8.93	0.45506
AVERAGE	28.05	24.8	0.4849

Table 4 with Decision tree, Table 5 with Random Forest Regressor, Table 6 with Gaussian Regression, Table 7 with Linear Regression, and Table 8 with Support Vector Regression. The neural network ends up predicting the stock price with its given data at a mediocre level at best, with MSE averages of 29.93 and 26.85 respectively. Its highest weight, tesla, ends up with only 0.013% of the total weightage. The MSE is relatively low on both Apple and Amazon stock but is high on the Tesla column, meaning that this model is not a good predictor of Tesla stock prices. The RANSAC Regression model performs incredibly well, with average MSE's of 28.05 and 24.8 MSE respectively. These MSE's don't seem to be that spectacular, so why is the Average weight so high? The Tesla MSE was significantly larger than that of Apple and Amazon, both of which were under ten for testing and training. Tesla ended up with a significantly larger relative MSE, with Apple and Amazon with MSE's under 10 each. Counterintuitively, however, this regression model still ends up with around a 0.49 weightage for Tesla stock, and similar Apple and Amazon weight values. This Model takes up around 50% of all the weightage from all 7 models, proving itself to be highly accurate.

Table 4. Decision tree

COMPANY	MEAN SQUARED ERROR (Testing)	MEAN SQUARED ERROR (Training)	WEIGHT
TESLA	155.74	0.00	$1.43e^{-62}$
APPLE	12.54	0.00	$1.056e^{-43}$
AMAZON	15.64	0.00	$1.044e^{-45}$
AVERAGE	61.306	0.00	$3.555e^{-44}$

Table 5. Random forest regressor

COMPANY	MEAN SQUARED ERROR (Testing)	MEAN SQUARED ERROR (Training)	WEIGHT
TESLA	97.98	10.17	$4.669e^{-26}$
APPLE	6.55	1.10	$3.733e^{-20}$
AMAZON	9.81	1.60	$1.127e^{-18}$
AVERAGE	38.11	4.29	$3.881e^{-19}$

Table 6. Gaussian regression

COMPANY	MEAN SQUARED ERROR (Testing)	MEAN SQUARED ERROR (Training)	WEIGHT
TESLA	57513.62	0.01	0
APPLE	12888.27	0.00	0
AMAZON	316160.08	0.00	0
AVERAGE	128853.99	0.003	0

Table 7. Linear regression

COMPANY	MEAN SQUARED ERROR (Testing)	MEAN SQUARED ERROR (Training)	WEIGHT
TESLA	71.50	59.17	0.49993
APPLE	4.80	6.30	0.4992
AMAZON	7.84	8.93	0.54413
AVERAGE	28.04	24.8	0.51466

Table 8. Support vector regression

COMPANY	MEAN SQUARED ERROR (Testing)	MEAN SQUARED ERROR (Training)	WEIGHT
TESLA	1045.28	633.43	$1.164e^{-162}$
APPLE	58.45	56.20	$1.056e^{-63}$
AMAZON	81.601	129.04	$2.449e^{-27}$
AVERAGE	395.11	272.89	$8.1633e^{-28}$

The Decision Tree, probably due to its simplicity, demonstrated subpar performance in the average Mean Squared Error (MSE) metric. While achieving a perfect 0.00 MSE for all three companies in its training data, it faltered in the relative MSE context, displaying poor performance.

This model accounts for less than 0.00000000000000000000000000000000000001% of the total weighting across all models. This model underwent "overfitting" which is when a model performs perfectly on its training data but unreliable when given new unrecognized data (such as the testing data). The Random Forest Regressor outperformed the Decision Tree in testing Mean Squared Error (MSE) but experienced an increase in training MSE. However, it garnered a notably higher weighting compared to the Decision Tree, representing less than 0.0000000000000001% of the cumulative weighting across all seven models. Performing the worst out of all seven models was the Gaussian regression model. This model acquired an average weightage of zero throughout all three of its trials, meaning that this model was inaccurate every single time. It also has a testing MSE higher than any other model, reaching 6 digits. Like decision tree, it victim to overfitting, making it an unworthy candidate for stock prediction. Like RANSAC Regression, Linear regression has incredible weightage and a mediocre average MSE. It seems all 7 models consistently underperform on Tesla, explaining the mediocre average SME for this model. Linear regression performs even better than the RANSAC

Regression model, controlling over half of the weightage. SVR (Support Vector Regression) doesn't seem to have a relatively weightage and has a high relative MSE, meaning that this model was very inaccurate in determining stock market prices. It controlled less than 0.00000000000000000000000000001% of the overall weightage.

DISCUSSION

Based on these results, it is determined that the Linear Regression (Table 7) and RANSAC regressor model (Table 3) both tend to have the highest weigh value for all 3 companies, and therefore are the most valuable when aiming to predict the market. Both models end up with very similar weight values and have the lowest average MSE for testing data, which explains why they also have the highest weight values (Most ML models tend to perform worse on testing data). When considering the weight values, it's important to note that all the values added together from every single model combined for each company are equal to 1. For example, when testing all of the 7 models for Apple, the value corresponding with weight for all the models should sum up to 1. The other 5 models ended up with extremely insignificant weight values, which explains why both the RANSAC regressor and the Linear Regression model each took up almost half the weightage each. There are differences in data between the RANSAC Regressor and the Linear regressor, of course, with the Linear Regression model holding an average of around 0.51 weightage and the RANSAC model holding an average of around 0.49 throughout all three companies. This would technically make the RANSAC model inferior to the Linear Regression model, but the differences are so vast between these models and the other 5 that we can refer to both the LR and RANSAC model as reliable. Other research conducted on the topic of using ML to unearth the highest performing models supports the conclusion of the neural network, which performed third out of all 7 models in terms of weightage. The study (Gupta, Tiwari, Bhatnagar, Shalu, Singh, & Ranjan, 2023) concluded that out of the two models, Random Forest Regressor, and a Neural Network model, the neural network models performs superiorly in predicting the stock market. This study takes the study a lot farther, examining 7 models instead of two to find those that perform even better than the neural network. The Gaussian model surprisingly performed extremely poorly, with an average mean squared error of 128853.99 for the testing data. This incredibly high MSE results in the Gaussian regressor holding a weightage of 0 for all 3 companies, meaning that it shouldn't even be considered when predicting stock prices and that the model is almost never accurate. This could be a result of the inability for a Gaussian model to plot sole linear data, and why the linear model itself performed superior to all other models.

Conclusion

This research thoroughly assessed seven predictive models, identifying the RANSAC Regressor and Linear Regression as superior tools for stock price prediction. The significance of simpler, linear models in capturing market trends underscores their reliability, albeit with recognition of inherent limitations when faced with unforeseen market dynamics. While this study's findings remain consistent across these models, their applicability may be dependent on specific dataset characteristics and timeframes (such as a linear upward trend). Validation using diverse datasets and varying market conditions would bolster the accuracy of these findings across broader financial landscapes. Future research endeavors should expand the scope of stock prediction models by integrating them with expansive language models (for example, ChatGPT). Incorporating these larger language models along with specific datasets will enable the inclusion of multifaceted factors into the prediction process such as decisions made by stock owners, market volatility, customer sentiments, and more. This research, taking solely market data into account, serves as a stepping-stone for further advancement of this type of research. The aim is to enhance predictive accuracy and comprehensiveness, gradually reducing the perceived unpredictability of stock markets. The integration of predictive models with advanced language-based AI and comprehensive datasets marks a pivotal stride toward unlocking deeper insights into stock market behavior. By integrating a diverse array of important, influential factors, future stock prediction models can evolve into more comprehensive and robust tools, offering a clearer understanding of market dynamics and contributing to a gradual reduction in the perceived unpredictability of stock markets. This research paper serves as an essential step towards harnessing the potential of broader stock prediction models. Future endeavors must continue this trajectory by expanding model capacities, incorporating richer data sources, and embracing AI-driven advancements to propel stock market predictability into new realms.

Conflicts of Interest: The authors declare no conflicts of interest.

References

Colah (2014). Neural Networks, Manifolds, and Topology. Colah's Blog. https://colah.github.io/posts/2014-03-NN-Manifolds-Topology/

Gupta, B. K., Tiwari, S., Bhatnagar, S., Shalu, Singh, Y., & Ranjan, T. (2023). Stock Price Prediction Using ML Algorithms. *International Journal for Research in Applied Science & Engineering Technology,* 11, 5426-5432. https://doi.org/10.22214/ijraset.2023.52944

Hardesty, L. (2017). Explained: Neural Networks. MIT News. https://www.thestreet.com/dictionary/o/opening-and-closing-prices

Investopedia (2023). What Is Closing Price? Definition, How It's Used, and Example. http://www.investopedia.com/terms/c/closingprice.asp

Klein, B. (2023). Neural Networks, Structure, Weights and Matrices. PythonCourse. https://python-course.eu/machine-learning/neural-networks-structure-weights-and-matrices.php

Kumar, A. (2020). RANSAC Regression Explained with Python Examples. Vitalflux. https://vitalflux.com/ransac-regression-explained-with-python-examples/

Macrotrends (2024). Stock Screener. https://www.macrotrends.net/stocks/stock-screener

Mwiti, D. (2023). Random Forest Regression: When Does It Fail and Why? Neptune.ai. https://neptune.ai/blog/random-forest-regression-when-does-it-fail-and-why

Sahai, K. N. (2023). Mastering Random Forest Regression: A Comprehensive Guide. Analytix Labs. https://www.analytixlabs.co.in/blog/random-forest-regression/

Sander, R. (2021). Gaussian Process Regression from First Principles. Towards Data Science. http://towardsdatascience.com/gaussian-process-regression-from-first-principles-833f4aa5f842

Singh, M. K. (2023). Support Vector Regression (SVR) Using Linear and Non-Linear Kernels in Scikit Learn. GeeksforGeeks. http://www.geeksforgeeks.org/support-vector-regression-svr-using-linear-and-non-linear-kernels-in-scikit-learn/

Wong, A., Figini, J., Raheem, A., Hains, G., Khmelevsky, Y., & Chu, P. C. (2023). Forecasting of Stock Prices Using Machine Learning Models. 2023 IEEE International Systems Conference (SysCon) (pp. 1-7). Institute of Electrical and Electronics Engineers. https://doi.org/10.1109/SysCon53073.2023.10131091

Xu, B., Kuplici, R., Sen, S., & Paulus, M.P. (2021). The Pitfalls of Using Gaussian Process Regression for Normative Modeling. Cold Spring Harbor Laboratory. https://doi.org/10.1101/2021.05.11.443565

****