HIGH SCHOOL EDITION
Journal of Student Research
Volume 13 Issue 1 (2024)

Using Decision Tree Regression and Related Companies' Stock Data to Predict Microsoft Stock Returns

Zachary Tong

Wheeler High School, USA

ABSTRACT

Using AI machine learning (ML) models to predict stocks is a topic that has already been studied by the stock market research community. However, two ML methods have not been analyzed in the current literature regarding stock prediction: the decision tree regression analysis and the related company training data approach. Thus, this study will utilize both of these unfamiliar stock prediction methods to predict Microsoft's stock returns. To begin with, the company data of Microsoft's biggest partners and competitors were imported from YahooFinance; this data was then used to form all the features for the stock prediction model (mean, standard deviation, price gaps, etc.). Next, the machine learning model was created using Python's Decision Tree Regressor method; the model was trained using data before 10/1/2001 and tested using data after 10/1/2001. Through repeatedly testing this model, hyperparameter tuning was performed to determine the model's best features and max depth for predicting Microsoft's stock returns. In the end, the final prediction model reached a percentage accuracy (percentage of times correctly predicting stock return's direction) of 56.68%, and the plot (net returns using model vs. historical net returns) showed that model use resulted in more consistent and significantly higher net Microsoft stock returns. Therefore, this study demonstrated that both the Decision Tree Regressor and the related company training data approach are successful machine learning methods in predicting Microsoft's stock returns. However, further research is required to extend this study's results to other companies and/or different stock metrics.

Introduction

In modern investors' eyes, historical data alone is not enough to sufficiently predict stock market trends (Khan et al., 2020). As a result, the current literature on stock market prediction has shifted its focus to machine learning models, using different types of ML methods to predict variations in stock metrics (Abe & Nakayama, 2018; Biswas et al., n.d.; Carta et al., 2020; Sheth & Shah, 2023; Umer et al., 2019). For instance, researchers studied the effectiveness of various stock price prediction models by testing 5 different ML methods: Long Short Term Memory, Extreme Gradient Boosting, Linear Regression, Moving Average, and Last Value (Biswas et al., n.d.). Similarly, a study conducted in 2011 compared the effectiveness of Artificial Neural Network, Support Vector Machine, and Long Short-Term Memory as ML methods in stock prediction (Sheth & Shah, 2023). On the other hand, multiple studies analyzed how Deep Learning could be implemented in successfully predicting variations in stock metrics (Abe & Nakayama, 2018; Carta et al., 2020), while a study in 2019 utilized the Linear Regression method of machine learning to forecast Amazon, Apple and Google stocks (Umer et al., 2019). From existing journals and studies, it is clear that many types of machine learning methods have been used in the stock market prediction field; however, not one of these studies tested a well-known ML method in Python: Decision Tree Regression (AnkanDas22, 2023). In fact, the only article in the current literature that examined a Decision Tree Regressor in relation to predicting stock price fluctuations was an unpublished, paper

presentation at an International Conference Proceeding (Karim et al., 2021). Therefore, the Decision Tree Regression ML stock prediction method exhibits an evident lack of study in the existing literature. However, to distinguish itself from even the unpublished conference proceedings, this study will use a Decision Tree Regressor to predict the variations in stock returns, not stock prices, of Microsoft, a widely known technological company that is still a strong buy in the current stock market community (Investor Express, 2023). Also, because the direction of stock metrics (positive or negative) is the most valuable factor in predicting stock trends (Kumar et al., 2011), this study's measurement of accuracy will be the percentage of times the Decision Tree Regressor correctly predicts the direction of Microsoft's stock returns.

Although this study's machine learning method and measurement of accuracy have all been determined, the specific data being used to train the model's stock prediction has not been established. Thus, training data methods for ML stock prediction in the current literature must be further examined. For starters, two journal articles studying non-machine learning methods of stock prediction used within-industry variables and aggregate output/rate as the primary data for their stock prediction models (Asness et al., 2000; Balvers et al., 1990). In contrast, researchers of a 2019 study analyzed the effects of adding the community's views and the political situation in ML stock prediction models (Khan et al., 2019). Two modern journals predicting market trends in Microsoft's stocks inputted Twitter data in machine learning models to successfully forecast movements in Microsoft stock prices (Koukaras et al., 2022; Vu et al., 2012). Lastly, researchers examining the impact of social variables on ML stock prediction found that both social media and financial news data significantly increased the accuracy of machine learning models' stock predictions (Khan et al., 2020). As a whole, the current literature has already extensively analyzed the success of using multiple stock metrics data to predict variations in stock market trends. However, no existing study nor journal has used related companies' stock data (companies' competitors and partners) as the primary data source for stock prediction models, that of which has already been shown to exhibit strong associations with stock trends (Harper, 2022). Therefore, this study will use the stock data of Microsoft's closely related companies to train the Decision Tree Regression model in accurately predicting Microsoft's stock returns.

Methods

Data Source

All the data used in this study came from public data from Microsoft and well-known technology companies associated with Microsoft. For starters, Microsoft's own company data was immediately inputted into the model, as this data was used as the base of the model. Next, the company data of Microsoft's biggest competitors and partners (Apple, Google, IBM, Samsung, SAP SE, Oracle, Fortinet, Salesforce, HubSpot, Adobe, and Managed Solutions) were all inputted into the ML stock prediction model; the independent and collaborative effects of each of these companies was then observed. In the end, only the competitors: Apple, Google, and IBM and the partners: Fortinet and HubSpot, were kept in the final model, for their cumulative effects on the model exceeded all other related company data combinations.

Retrieving the Data

All data was imported from YahooFinance using the yFinance package. A dataframe named *crypto_data* was created (using the pandas class) to store both Microsoft's independent company data as well as the individual data of all the related companies. Then, a new data frame was created for each individual company with an identifiable name to store and later access the databases of each company (e.g., dataframe named *ibm* stored all

of IBM's stock data). For all the companies used in this model, data was retrieved from the date, January 1st, 2020, up until the most recently accessible data.

Variables

Variables Directly from Data

The first variables created to test in the stock prediction model were retrieved directly from the company databases imported from YFinance. These variables included Opening Price, Closing Price, Adjusted Closing Price, Volume, Daily High Price, and Daily Low Price. For each of these company databases, these basic variables were labeled respectively and tested in the Decision Tree Regression model.

Created Variables

Each company database also had a set of variables created by combining multiple existing variables in the company database. For instance, Yesterday's Return for each company stock was calculated by dividing the Adjacent Close by the Adjacent Close a day earlier and then subtracting one (Piepenbreier, 2022). The Mean and Standard Deviation of the Return over the last 10 and 20 days were calculated by finding the rolling mean and std of the return and shifting by 1. Some other created variables included the Mean of the Last 10 Days Weighted, Yesterday's Volume, Mean Volume of Last 10 Days, Mean Volume of Last 20 Days, and Price Gaps (calculated by subtracting Opening Price shifted by one from Closing Price shifted by one). Every created variable was applied to each company and labeled in correspondence with the respective company database.

Combination Variables

Over the course of this study, hundreds of combination variables were formed by combining basic and created variables, but after extensive hyperparameter tuning, only 11 of these combination variables proved beneficial to the stock predictor model. These variables included: 1) Microsoft Mean Last 10 Days divided by Microsoft Mean Last 20 Days; 2) Apple Mean Last 10 Days divided by Apple Mean Last 20 Days; 3) Microsoft Mean Last 10 Days divided by Microsoft Yesterday's Volume; 4) Microsoft Price Gaps divided by Microsoft Yesterday's Volume; 5) IBM Mean Last 10 Days divided by IBM Mean Last 20 Days; 6) IBM Standard Deviation Last 10 Days divided by IBM Standard Deviation Last 20 days; 7) Fortinet Standard Deviation Last 10 Days divided by Fortinet Standard Deviation Last 20 days; 8) Fortinet Yesterday Return divided by Fortinet Mean Last 10 Days; 9) Fortinet Mean Last 10 Days divided by Fortinet Yesterday Volume; 10) Fortinet Price Gaps divided by Fortinet Yesterday Volume; and 11) HubSpot Mean Last 10 Days divided by HubSpot Yesterday Volume.

Model

Preparing Model

To form the stock prediction model, new databases for training and testing the model had to be created. First, a new database named *for_model* was formed to store all the features in *crypto_data* converted to datetime. Next, I created *training* and *testing* databases, where the *training* database would be used to fit (train) the prediction model while the *testing* database would be used to make (test) the actual predictions. The *training* database contained all the data in *for_model* before October 1st, 2021, while the *testing* database contained all the data in *for_model* after October 1st, 2021. Because the independent and dependent variables must be separated to train and predict a Decision Tree Regressor, both the *training* and *testing* databases were divided into *x_train,*

y_train, x_test, and *y_test. X_train and x_test* contained the features from *crypto_data* that will be used to make the predictions, while *y_train* and *y_test* contained only Microsoft's stock return (the variable being predicted).

Creating Model

A Decision Tree Regressor was used as the Machine-Learning model for this project. To create this model, a new variable named *model* was created, and it was set to the DecisionTreeRegressor method with random_state equal to 0. Next, the fit method of Sklearn was used to train the machine learning model, *model,* using data in *for_model* prior to October 1st, 2001 (with *x_train* as the independent variable and *y_train* as the dependent variable). The actual Microsoft stock return predictions, however, were made by using the predict method of Sklearn on *model* and incorporating company data after October 1st, 2001 (with *x_test* as the only parameter and independent variable). Finally, all the Decision Tree Regressor's predictions were then assembled into a database named *predicted.*

Testing Model

To test the model, the resulting *predicted* database from earlier was added into the *testing* database under a column named "predicted". Another column in the *testing* database was created named "correct", and this column was the multiplicative product of the Microsoft Return and Predicted columns of the *testing* database. Because this multiplicative product represents stock return direction accuracy (positive means correctly predicted direction; negative means incorrectly predicted direction), I modified the *testing* database to only include the data in the "correct" column that were greater than 0 (indicating correct direction predictions). I then divided the length of that resulting column by the length of the original "correct" column to find the percentage of correct direction predictions made by the Decision Tree Regressor. Lastly, the net Microsoft returns using the prediction model was plotted against the historical net Microsoft returns to visually observe the effects of the created stock prediction model. The upward consistency of the plot and the value of the percentage were the primary testing statistics used for both testing the machine learning model's accuracy and hyperparameter tuning the max_depth and features of the model.

Results

After performing hyperparameter tuning by experimenting with different max_depths and model features, the Decision Tree Regressor with max_depth of 35 ended up producing the most accurate and consistent Microsoft stock return predictions. Furthermore, the most beneficial features kept in the final model were Microsoft Mean Last 10 Days, Microsoft Mean Last 20 Days, Microsoft Standard Deviation Last 10 Days, Microsoft Mean Last 10 Days Weighted, Microsoft Mean Volume Last 20 Days, Apple Mean Last 10 Days, Apple Mean Last 20 Days, Apple Mean Volume Last 10 Days, Apple Mean Volume Last 20 Days, Google Mean Volume Last 20 Days, IBM Mean Last 20 Days, and all 11 combination variables discussed earlier by combining basic/advanced variables. Using the determined max_depth and final features, the model's highest percentage of correctly predicting Microsoft's return directions was 56.68%, although the percentage varies on a day-to-day basis due to the constantly changing data. Furthermore, the plot was quite consistent despite the changing data, with the Microsoft net returns using the ML stock prediction model increasing relatively constantly and consistently being higher than the historical Microsoft net returns. In general, the Microsoft net returns using the prediction model would reach $1.75 per share by October 2023, whereas the historical Microsoft net returns would reach around $0.25 per share by October 2023.

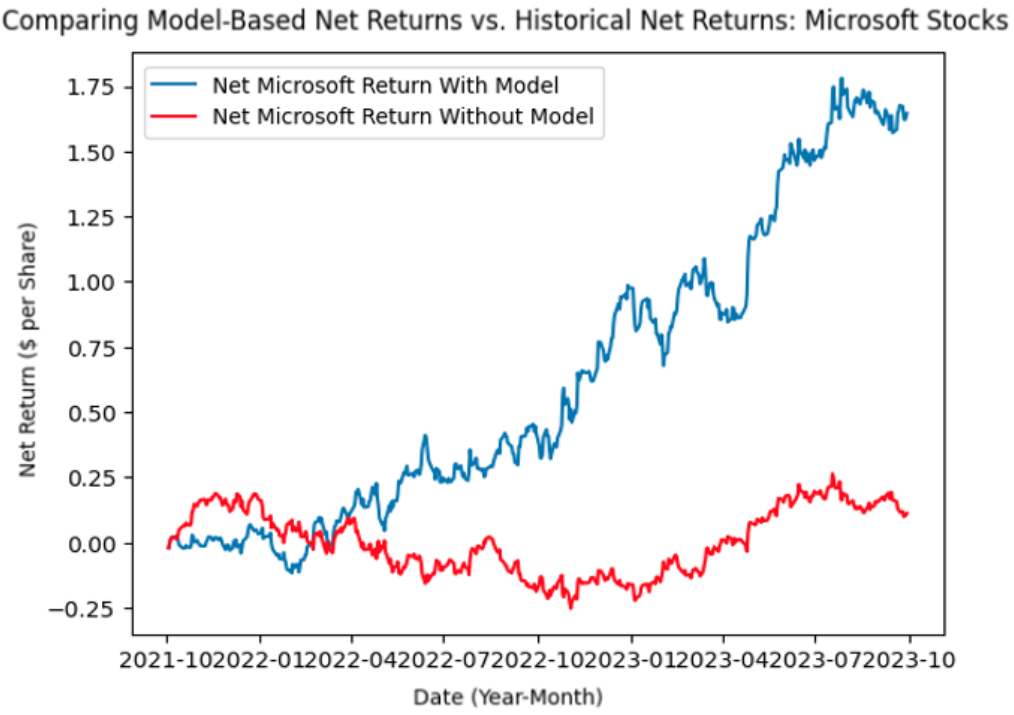

Figure 1. Comparing model-based net returns vs. historical net returns: Microsoft stocks

Discussion

In the end, the final direction-accuracy percentage of the Decision Tree Regressor model using Microsoft's related company data was lower than the stock prediction models using Twitter data (Koukaras et al., 2022; Vu et al., 2012). However, the Decision Tree Regression method and the related company training data approach still proved to be successful in predicting Microsoft's stock returns, as this study proved that the use of these ML stock prediction methods resulted in a direction-accuracy percentage safely over 50%, indicating a clear increase from the base assumption odds. Additionally, the plot of the Decision Tree Regressor's Microsoft net returns vs the historical Microsoft net returns demonstrated that both the Decision Tree Regressor and related company model-training data significantly increased Microsoft's net returns by over $1.50 per share in 2 years. Therefore, because both the Decision Tree Regression method and related company training data approach were previously unfamiliar to the current literature on ML stock prediction models, through these results, a new machine learning method and a new model-training data approach have been established as successful methods in predicting a certain stock metric (Microsoft's stock returns).

Conclusion

Through this study, both the Decision Tree Regressor and the related company training data approach were shown to be effective methods in predicting Microsoft's stock returns. In the future, further studies are needed to apply the machine learning approaches to stock prediction of other large-scale companies and/or different stock metrics. That way, the results of these studies can be extended beyond just Microsoft's stock returns, and more conclusions can be drawn about the effectiveness of these unfamiliar methods in ML stock prediction. Additionally, future studies should also consider combining these approaches with other machine learning

methods, as these new stock prediction methods may prove even more successful in collaboration with other proficient methods in the existing literature.

Limitations

In this study, the Decision Tree Regressor and related company data training procedures were both tested at the same time, so the results of this study cannot distinguish the effects of each approach on the stock prediction model's accuracy. Furthermore, this study used the Decision Tree Regressor and related company training data approach to only predict Microsoft's stock returns. As a result, this study alone cannot ascertain that these methods will produce the same results when applied to companies outside of Microsoft or stock metrics other than stock returns. Lastly, this study used the Decision Tree Regressor and related company training data approaches to independently predict Microsoft returns without using any other data or machine learning models. Thus, the collaborative effects of incorporating these new-to-the-literature methods into other stock prediction models in the current literature cannot be determined.

Acknowledgments

I would like to thank Dr. Aliya Babul at Columbia University for assisting me in the Python programming aspects of this study's machine-learning stock prediction model. I would also like to thank Christopher Cahill for reviewing the first draft of this research manuscript.

References

Abe, M., & Nakayama, H. (2018). Deep learning for forecasting stock returns in the cross-section. *Advances in Knowledge Discovery and Data Mining*, 273-284. https://doi.org/10.1007/978-3-319-93034-3_22

AnkanDas22. (2023, January 11). *Python | decision tree regression using sklearn*. Geeks for Geeks. https://www.geeksforgeeks.org/python-decision-tree-regression-using-sklearn/#
Asness, C. S., Porter, R. B., & Stevens, R. L. (2000). Predicting stock returns using industry-relative firm characteristics. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.213872

Balvers, R. J., Cosimano, T. F., & McDonald, B. (1990). Predicting stock returns in an efficient market. *The Journal of Finance*, *45*(4), 1109-1128. https://doi.org/10.1111/j.1540-6261.1990.tb02429.x

Biswas, M., Shome, A., Islam, M. A., Nova, A. J., & Ahmed, S. (n.d.). Predicting stock market price: A logical strategy using deep learning. *Institute of Electrical and Electronics Engineers*. https://doi.org/10.1109/iscaie51753.2021.9431817

Carta, S., Corriga, A., Ferreira, A., Podda, A. S., & Recupero, D. R. (2020). A multi-layer and multi-ensemble stock trader using deep learning and deep reinforcement learning. *Applied Intelligence*, *51*(2), 889-905. https://doi.org/10.1007/s10489-020-01839-5

Harper, D. R. (2022, July 22). *Forces that move stock prices* (T. Brock & K. R. Schmitt, Ed.). Investopedia. https://www.investopedia.com/articles/basics/04/100804.asp#:~:text=Company%20stocks%20tend%20to%20track%20with%20the%20market,individual%20performance%E2%80%94determines%20a%20majority%20of%20a%20stock%27s%20movement.

Investor Express. (2023, September 26). Microsoft: 5 reasons why the stock is a strong buy. https://seekingalpha.com/article/4637510-microsoft-5-reasons-why-the-stock-is-a-strong-buy

Karim, R., Alam, K., & Hossain, R. (2021, August 10). *Stock market analysis using linear regression and decision tree regression* [Paper presentation]. 2021 1st International Conference on Emerging Smart Technologies and Applications, Lebanese International University, Sana'a, Yemen. https://ieeexplore.ieee.org/abstract/document/9515762

Khan, W., Ghazanfar, M. A., Azam, M. A., Karami, A., Alyoubi, K. H., & Alfakeeh, A. S. (2020). Stock market prediction using machine learning classifiers and social media news. *Journal of Ambient Intelligence and Humanized Computing*, *13*(7), 3433-3456. https://doi.org/10.1007/s12652-020-01839-w

Khan, W., Malik, U., Ghazanfar, M. A., Azam, M. A., Alyoubi, K. H., & Alfakeeh, A. S. (2019). Predicting stock market trends using machine learning algorithms via public sentiment and political situation analysis. *Soft Computing*, *24*(15), 11019-11043. https://doi.org/10.1007/s00500-019-04347-y

Koukaras, P., Nousi, C., & Tjortjis, C. (2022). Stock market prediction using microblogging sentiment analysis and machine learning. *Telecom*, *3*(2), 358-378. https://doi.org/10.3390/telecom3020019

Kumar, L., Pandey, A., Srivastava, S., & Darbari, M. (2011). A hybrid machine learning system for stock market forecasting. *Journal of International Technology and Information Management*, *20*(1). https://doi.org/10.58729/1941-6679.1099

Piepenbreier, N. (2022, June 3). *How to add titles to matplotlib: Title, subtitle, axis titles*. Datagy. Retrieved October 7, 2023, from https://datagy.io/matplotlib-title/

Sheth, D., & Shah, M. (2023). Predicting stock market using machine learning: Best and accurate way to know future stock prices. *International Journal of System Assurance Engineering and Management*, *14*(1), 1-18. https://doi.org/10.1007/s13198-022-01811-1

Umer, M., Awais, M., & Muzammul, M. (2019). Stock market prediction using machine learning(ml)algorithms. *ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal*, *8*(4), 97-116. https://doi.org/10.14201/adcaij20198497116

Vu, T. T., Chang, S., Ha, Q. T., & Collier, N. (2012, December). An experiment in integrating sentiment features for tech stock prediction in twitter. In *Proceedings of the workshop on information extraction and entity analytics on social media data* (pp. 23-38). https://aclanthology.org/W12-5503.pdf