Forecasting stock price trends with machine learning techniques

Tianyao Li

Department of Computer Science, Florida International University, Miami Florida, USA

tli027@fiu.edu

Abstract. The stock market is a landscape of risk and unpredictability, where one wrong move can result in substantial financial loss. Hence, predicting stock market movements becomes paramount for informed investing. This research harnesses the power of four unique machine learning algorithms: Decision Tree, Linear, Random Forest, and Support Vector Regressions, aiming to forecast Apple Inc.'s stock prices. Historical data spanning three years served as the foundation for training these predictive algorithms. These models were subsequently evaluated and juxtaposed using the mean squared error metric, a robust measure of predictive accuracy. The analysis discerned that the Support Vector Linear Regression exhibited superior performance relative to its counterparts, shedding invaluable light for stock investors navigating the intricate financial markets. This evaluation of models also reveals potential disparities in forecasting precision, underscoring the imperative of model selection. The insights garnered from this study not only provide direct implications for shaping investment strategies but also lay a solid groundwork for further explorations, particularly in the realm of advanced predictive techniques.

Keywords: Machine Learning, Stock Price, Forecast.

1. Introduction

As the pace of economic globalization quickens, so does the trend towards financial globalization, directing an ever-growing number of investors to the stock market. This makes predicting stock market behavior critical for investment and financial planning. However, the stock market is a labyrinthine and expansive system fraught with numerous variables and shifts, making stock price movements erratic and challenging to forecast. Machine Learning is emerging as a transformative force in stock price forecasting, with extensive research ongoing to develop models that can predict next-day closing prices for stocks, facilitating prudent investment decisions and potentially rewarding returns.

There is a wealth of existing research concerning stock price prediction in the financial markets, utilizing machine learning algorithms. Studies by Yoo et al. and others suggest that neural networks outperform some commonly-used prediction methodologies in terms of accuracy. Further work by Pahwa et al. evaluates the pros and cons of several prevalent machine learning algorithms and platforms including Linear Regression and Support Vector Machines. Another study by Usmani et al. develops new prediction methods using various Artificial Neural Network-based models and SVM, showing that Multi-Layer Perceptrons performed quite well. In another development, Parmar et al. assessed the efficacy of Long Short-Term Memory models in comparison to Regression-based models, finding the former to perform better in terms of accuracy.

Despite the extensive body of work, there's a gap in research specifically targeting high-tech corporations' stock price predictions. This paper fills that void by focusing on Apple Inc., employing multiple machine learning algorithms, including Linear Regression, Decision Tree Regression, Support Vector Regression, and Random Forest Regression, to make the forecast. The research calculated the mean squared errors for each model to evaluate their efficacy, concluding that linear models, specifically Support Vector Linear Regression, showed the best performance.

2. Data and methods

2.1. Data analysis

For this research centered focus on Apple Inc.'s stock data. Apple, being a leading technology giant, boasts a staggering market capitalization of 2.9 trillion dollars. The time frame for data collection extends from March 4th, 2019 to March 2nd, 2022, and the data was procured from Yahoo Finance (https://hk.finance.yahoo.com/). Initially, dataset contained 7 distinct columns, each having a consistent 756 entries, and notably, there was an absence of any missing values. One of initial challenges was the "Date" column, which was registered as an "object" type. The research promptly converted this to "datetime64" on a cloned dataset to better suit subsequent analytical endeavors. The next steps in preprocessing included the identification and imputation of missing values, where gaps were filled using either median or arithmetic average of the respective columns.

There is significant price volatility observed over the selected time frame. The average closing price is slightly above 100, with the prices fluctuating between a low of 43.125 and a high of 182.010. With the overarching aim to predict the closing price, the study focused on the 'Close' data and shifted its attention to the 'Date' data, preparing it for predictive analysis.

2.2. Methodology

2.2.1. Linear regression. Linear Regression operates on the foundational assumption that there exists an approximate linear correlation between the variables X and Y. This interrelation can be numerically expressed using two pivotal parameters, namely, β_0 and β_1. These represent the y-intercept and gradient in the linear equation. The elementary linear model is encapsulated as [1]:

$$Y \approx \beta_0 + \beta_1 X. \quad (1)$$

For the stock data set, the research employed the “date” as the target and aimed to predict the “close” feature. Upon calibrating model using historical data, which provided estimates for β_0 and β_1, the research was equipped to forecast subsequent closing prices rooted in historical date values [2].

$$y = \hat{\beta}_0 + \hat{\beta}_1 x. \quad (2)$$

2.2.2. Decision trees in stock forecasting. In the research, Decision Trees serve as a valuable tool for stock price prediction, offering several benefits as outlined in equation, primarily, it acts as a robust exploratory tool. Decision Trees, with their intrinsic nature, provide a relatively quicker avenue to validate significant factors. Furthermore, these trees can effortlessly pinpoint latent variables. An added benefit is their ability to recognize non-linear patterns, which proves invaluable for stock price predictions.

Decision Trees have earned their reputation as one of the most prevalent machine learning techniques, finding use in competitive scenarios like Kaggle as well as in industrial applications. In essence, a decision tree operates by posing a series of binary questions until a particular forecast is ascertained. It autonomously formulates the sequence and essence of these queries.

Decision Trees have versatility, catering to both classification and regression challenges. The output distinction leads to regression trees or classification trees. However, their construction remains analogous, comprising elements like the Root, Node, and Leaf. Notably, a regression tree yields a continuous value, differing from a class-based outcome.

The foundation of a Decision Tree revolves around the concept of information gain. Given a dataset D, where $|D|$ denotes its sample magnitude, there exist K unique categories, denoted as Ck (k ranging from 1 to K). The size of samples corresponding to C_k is $|\mathrm{C}_k|$, with the condition: $\sum_{k=1}^{K}|C_k| = |D|$. Assuming feature A exhibits n unique values$\{a1, a2, \ldots, an\}$, it segments D into subsets: $D1, D2, \ldots, Dn$. The magnitude of sample D_i is $|D_i|$, maintaining the condition $\sum_{i=1}^{n}|D_i| = |D|$ equating to $|DD|$. D_{ik} is the intersection of samples in set D_i corresponding to category C_k, i.e., $D_{ik} = D_i \cap C_k$ with $|\mathrm{D}_{ik}|$ signifying the sample magnitude of D_{ik}. The methodology for determining information gain encompasses:

Evaluating the empirical entropy, $H(D)$, of dataset D [3]:

$$H(D) = -\sum_{k=1}^{K}\frac{|C_k|}{|D|}\log_2\frac{|C_k|}{|D|}. \tag{3}$$

Estimating the conditional entropy of D predicated on A [4]:

$$H(D \mid A) = \sum_{i=1}^{n}\frac{|D_i|}{|D|}H(D_i). \tag{4}$$

Deriving information gain [5]:

$$g(D, A) = H(D) - H(D \mid A). \tag{5}$$

2.2.3. The random forest methodology. Random Forest Regression encompasses multiple regression trees. These trees are essentially criteria or rules that are hierarchically arranged and sequentially applied. The methodology initiates with a randomly chosen sample, undergoing alterations within its training set. Each progression is mapped out by a designated regression tree. Nodes within every tree scrutinize the input variables, chosen indiscriminately from the dataset. Historically speaking, random forests tend to outperform decision trees in forecasting accuracy. However, both models appear inferior when compared to gradient-boosted trees. As with numerous forecasting models, the chosen sample data exerts a profound influence on the accuracy of predictions.

In training trees, Random Forest uses bootstrap aggregation, often termed as 'bagging'. Given a training dataset represented by $X = x_1, \ldots, x_n$ and their corresponding outputs Y $= y_1, \ldots, y_n$, bagging operates by selecting a sample arbitrarily. For each value of b in the range 1 to B.

a. It selects n training instances from X and Y, denoting them as X_b and Y_b;

b. A classification or regression tree, represented as fb, is then established on X_b, Y_b.

Upon completion of the training phase, predictions for unknown samples can be determined by averaging the forecasts from all distinct regression trees on x' [6]:

$$\hat{f} = \frac{1}{B}\sum_{b=1}^{B} f_b(x'). \tag{6}$$

Moreover, to assess the variability in predictions, the standard deviation of forecasts across all distinct regression trees on x' can be computed as [7]:

$$\sigma = \sqrt{\frac{\sum_{b=1}^{B}(f_b(x') - \hat{f})^2}{B-1}}. \tag{7}$$

B denotes the number of samples or trees, acting as a variable parameter. It's commonplace to employ hundreds to even thousands of trees, contingent on the nature and size of the training dataset. The optimal count of trees can be ascertained via the cross-validation error, essentially the median prediction error for every training instance xi. Both training and test errors tend to plateau once a specific number of trees have been constructed and adjusted.

2.2.4. Exploring support vector methods. Support Vector Regression (SVR) stands as a versatile learning mechanism designed to tackle challenges related to function approximation. Within its procedural framework, the Structural Risk is earmarked for optimization with the intent of reduction. Using training instances represented as $\{(\mathrm{x}_1, y_1), \ldots, (\mathrm{x}_l, y_l)\}$ each x_i from the set R^d acts as an input training vector. On the other hand, y_i from the set R^1 signifies a desired output. Comprehensive mathematical representations related to this are provided hereunder [8, 9, 10, 11, 12, 13]:

$$\min_{\mathrm{w},b,\xi,\xi^*} \frac{1}{2}\mathrm{w}^T\mathrm{w} + C\sum_{i=1}^{l}\xi_i + C\sum_{i=1}^{l}\xi_i^*, \tag{8}$$

$$y_i - \mathrm{w}^T\phi(\mathrm{x}_i) - b \leq \varepsilon + \xi_i, \tag{9}$$

$$\&\mathrm{w}^T\phi(\mathrm{x}_i) + b - y_i \leq \varepsilon + \xi_i^*, \tag{10}$$

$$\xi_i, \xi_i^* \geq 0, i = 1, \ldots, l, \tag{11}$$

$$\min_{\alpha,\alpha^*} \frac{1}{2}(\alpha - \alpha^*)^T\mathrm{Q}(\alpha - \alpha^*) + \varepsilon\sum_{i=1}^{l}(\alpha_i - \alpha_i^*) + \sum_{i=1}^{l} y_i(\alpha_i - \alpha_i^*), \tag{12}$$

$$\sum_{i=1}^{l}(\alpha_i - \alpha_i^*) = 0, 0 \leq \alpha_i, \alpha_i^* \leq C, i = 1, \ldots, l. \tag{13}$$

Translating the essence of these equations, the vectors X_i shift from a lesser dimensional space to a comparatively elevated, potentially infinite, dimensional attribute domain. The parameter 'C' orchestrates the equilibrium between f's continuity and the allowance for grade deviations surpassing ε. The parameter ε is explicated in equation (14) embodying the criterion for Vapnik's ε-insensitive loss function [14, 15]:

$$|\mathrm{x}|\varepsilon = max(0, |\mathrm{x}| - \varepsilon). \tag{14}$$

SVR's functional equation is delineated as:

$$f(\mathbf{x}) = \sum_{i=l}^{l}(-\alpha_i + \alpha_i^*)K(\mathbf{x}_i, \mathbf{x}) + b \tag{15}$$

A vast array of kernel functions, instrumental in transmuting raw data into a suitable format for processing, exists to offer flexibility in handling diverse regression scenarios.

3. Conclusions from the data

Once models are fine-tuned, they're subjected to evaluation using fresh datasets. Their forecasting abilities are graphically represented, typically through line diagrams. The Mean Squared Error (MSE) metric provides an insight into each model's precision. By juxtaposing these MSE values, one can deduce which among the four models excels in performance.

3.1. Analyzing predictive models

The analysis involves assessing the accuracy of four distinct models in estimating closing prices. The data reveals that the Linear Regression and SVR (employing a Linear Kernel) predictions, although slightly undervalued, generally align with actual figures, except during unexpected fluctuations. On the contrary, the SVR model with a Polynomial Kernel appears to deviate significantly in its trend predictions. Furthermore, the Decision Tree and Random Forest Regressors also produce estimates that vary considerably from the actual data [16].

3.2. Analyzing model discrepancies

While the models previously discussed provide insights into the forecasted values, a tangible metric for precise evaluation is still required. The Mean Squared Error (MSE) is selected as the evaluation benchmark in this study. The data collected clearly indicates that the Support Vector Regressor, when combined with a linear kernel, yields the most notable MSE among the models compared.

Different models showcased varying MSE values. The MSE values for Linear Regression, Decision Tree Regression, Support Vector Regression (both linear and polynomial kernels), and Random Forest Regression are 126.907, 241.585, 114.442, 722.522, and 128.008 respectively. Of these, the Support Vector Regressor paired with the linear kernel delivered the lowest MSE at 114.442 [17].

4. Conclusion

This research incorporated diverse machine learning techniques, namely Decision Tree, Linear Regression, Support Vector Regression, and Random Forest Regression, for forecasting. Post analysis of the MSE for all models, it was discerned that both Linear Regression and linear-kernel SVR outshone their counterparts, offering fairly precise estimations about Apple's stock trajectory. Conversely, the alternate two models were subpar in their predictions. Such findings underscore the efficacy of linear algorithms for stock market forecasting.

In essence, this study serves as a valuable guide for market participants, albeit with certain limitations. While the linear-kernel SVR recorded the least error, it doesn't necessarily imply it's the most suited for all forecasting contexts. Both Random Forest and polynomial-kernel SVR models might benefit from fine-tuning. Despite its minor shortcomings, this research extends valuable insights to stock market investors. In prospective studies, a broader array of models could be examined against the current linear ones.

References

[1] Biswas M, Nova A J, Mahbub M K, Chaki S, Ahmed S, Islam M A 2021 ICSCT 1-6
[2] Sharma A, Bhuriya D, Singh U 2017 ICECA 2 506-509
[3] Yoo P D, Kim M H, Jan T 2005 CIMCA-IAWTIC 06 2 835-841
[4] Pahwa N, Khalfay N, Soni V, Vora D 2017 International Journal of Computer Applications 163 5 36-43
[5] Usmani M, Adil S H, Raza K, Ali S S A 2016 ICCOINS 322-327)
[6] Parmar I, Agarwal N, Saxena S, Arora R, Gupta S, Dhiman H, Chouhan L 2018 ICSCCC 574-576
[7] Kim Y S 2008 Expert Systems with Applications 34 2 1227-1234
[8] Ghosh A, Maiti R 2021 Environmental Earth Sciences 80 8 1-16
[9] James G, Witten D, Hastie T, Tibshirani R 2013 An introduction to statistical learning 61
[10] Drakos G 2019 Decision Tree Regressor Explained in Depth
[11] Hindrayani K M, Fahrudin T M, Aji R P, Safitri E M 2020 ISRITI 344-347
[12] Zhou X, Zhu X, Dong Z, Guo W 2016 The Crop Journal 4 3 212-219
[13] Piryonesi S M, El-Diraby T E 2020 Journal of Transportation Engineering 146 2 04020022
[14] Rabe A, van der Linden S, Hostert P 2009 In 2009 First Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing 1-4
[15] Chang C C, Lin C J 2011 TIST 2 3 1-27
[16] Sen J, Mehtab S 2019 Design and Analysis of Robust Deep Learning Models for Stock Price Prediction
[17] Liu L, Peng B, Yu J 2022 ICEDBC 2022