Stock price prediction based on multiple linear regression

Qi Wang[1, †], Chang Xu[2, *, †] and Tieyan Zhou[3, †]

[1]Beijing Haidian International School Beijing, China

[2]Hangzhou Ren He Experimental School Hangzhou, China

[3]New Channel Qingdao, China

*Corresponding author: 202011010111006@zcmu.edu.cn

[†]These authors contributed equally

Abstract. Stock price prediction plays an important role in finance and economics. In general, a rise and fall in the share price influences the investors' determinations and spurs the interest of the researchers over the years. The existing forecasting methods make usage of both linear and non-linear algorithms. From the share price fluctuating of NVDA, AMD, and INTC, we adopted the model MLR (multiple linear regression) to forecast the stock trend and find the relatives of the three stocks within the fixed period. Aside from this, correlation analysis was carried out, and several indexes and metrics were applied to evaluate the models. According to the analysis, three models are constructed, and two of them are relatively significant after improving. Overall, these results shed light on guiding further explorations of stock price forecasting based on the state-of-art financial and statistical models in the concept of big data analysis.

Keywords: Stock market; multiple linear regression; price prediction.

1. Introduction

The stock market leads the trend of the capital and shows the barometer of the markets. Investors try to follow their clues or method to predict or determine their investment goal, which raises the interest of the researchers or investors who would like to play a big role in their area. On this basis, the prediction process of the stock is an important part, which is also regarded as a difficult and complex part [1]. Because it requires mixing in the use of various factors and the special activity of individual factors (e.g., political, economic, market factors as well as technology and investor behavior), it will all result in changes in stock prices [2]. However, there are still some methods that could be used to predict the stock price approximately.

The existing methods for stock price prediction can be classified as follows: fundamental analysis, technical analysis, and time series prediction. Fundamental analysis is a type of investment analysis normally analyzed by sales, earnings, profits, and other economic factors; the technical analysis uses the historical price of stocks to identify the future price. Regarding the time series prediction, it involves basically two classes of algorithms: linear models and non-linear models. The different linear models are AR, ARMA, ARIMA, and their variations. Non-linear models involve methods that include ARCH, GARCH TAR, and deep learning algorithms [3].

Contemporarily, many people predict the stock in the market. In addition, they use different ways to calculate or estimate the data. With this in mind, a brief summary of previous results is reviewed primarily.

Islam and Nguye introduce the basis of machine learning: the KNN algorithm, decision tree, random forest, SVM (support vector machine), and linear regression. Then, they obtain the result through a single algorithm and compare each data to conclude respectively. According to the analysis, if the value of the model is closest to zero, the model will be suitable to predict stock, and the linear regression showed the best model midst KNN, SVM, DT, and RF, and these models are behind the SVM [4]. In another paper, researchers want to calculate different errors through APE (absolute percentage error), AAE (average absolute error), ARPE (average relative percentage error), and RMSE (root-mean-square error) to compare three models which are ARIMA, ANN, and Geometric Brownian Motion to predict the data. Finally, they concluded that ARIMA and Geometric Brownian

Motion model is better than the ANN models [5]. In addition, Vijh et al. collect data and construct the model to fit the data and evaluate the corresponding model. To be specific, six new variables models are built, and the random forest and ANN model are constructed to evaluate the effectiveness with the calculation of the values of RMSE, MAPE, and MBE. The result shows that the ANN includes the best value [6]. Besides, scholars also demonstrate the combination between DT (decision tree) and ANN model (prediction model) and DT and DT. They combine DT and DT, and ANN and DT. Moreover, they find the accuracy of DT+DT and ANN+DT, respectively. As a result, DT + ANN model has 77 percent accuracy [7].

In general, the main way to predict the stock is through machine learning and comparing with different models to predict the stock more easily after reading these papers. In this essay, based on the Stockstats library from python, we obtain the index of the data and predict the price trend of the three stocks via multiple linear regression. Based on a multiple linear regression model to predict the price of the stock and using coefficient analysis to improve the model are two priorities in this essay.

During the experiment, the daily close price of each stock was collected from a dataset. Subsequently, indexes of this dataset were calculated as the independent variable. Multiple linear regression was made to predict the close price for each stock. Finally, correlation analysis and evaluation metrics were made to improve the accuracy of the model better.

The rest of the essay is organized as follows. In Section II, datasets are first presented and used to build the multiple linear regression models. In addition, the evaluations would be done to test these models. In Section III, the result of the model and the limitation of the experiment will be shown and discussed. In Section IV, a conclusion will be made, and the limitation and the thesis will be further discussed.

2. Methodology

2.1 Data

Three stocks are chosen to be used as a prediction. They are NVDA, AMD, and INTC. Data are collected from www.investing.com. All three stocks' close prices are ordered by date from 1-Apr-20 to 29-Apr-22. The sample size is 525 rows. There is no outlier and gap in the datasets. The datasets contain the date, close, open, high, low, volume, and change. Figures 1~3 display the close price for Intel, AMD, and NVIDIA, respectively.

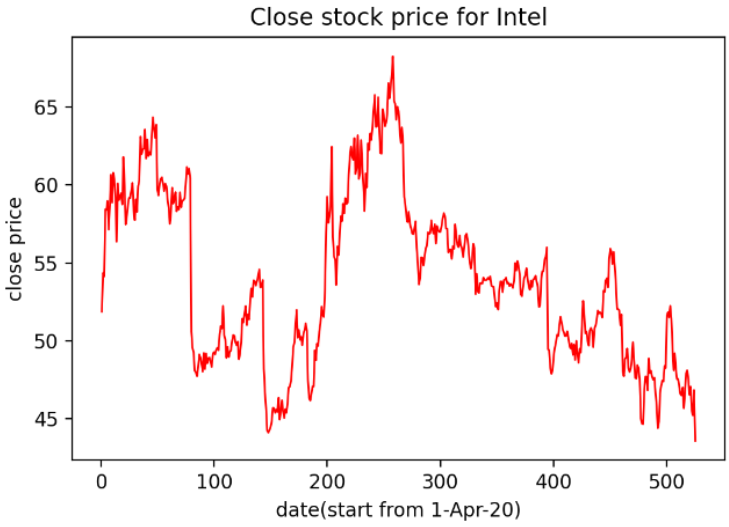

Figure 1. Close stock price of Intel in dollar per day from 1-Apr-2020.

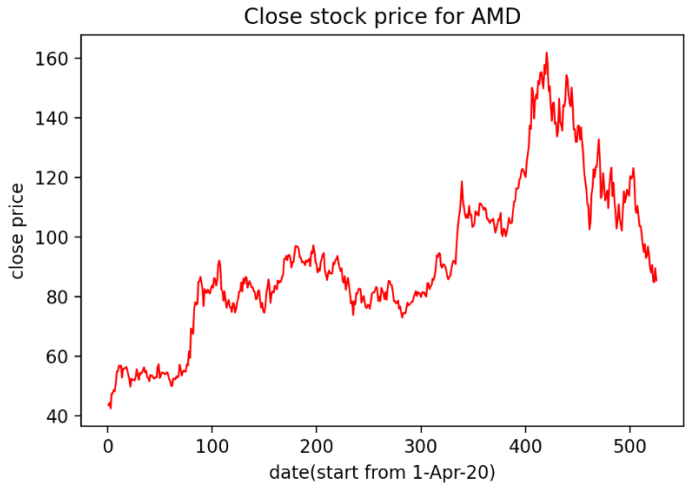

Figure 2. Close stock price of AMD in dollar per day from 1-Apr-2020.

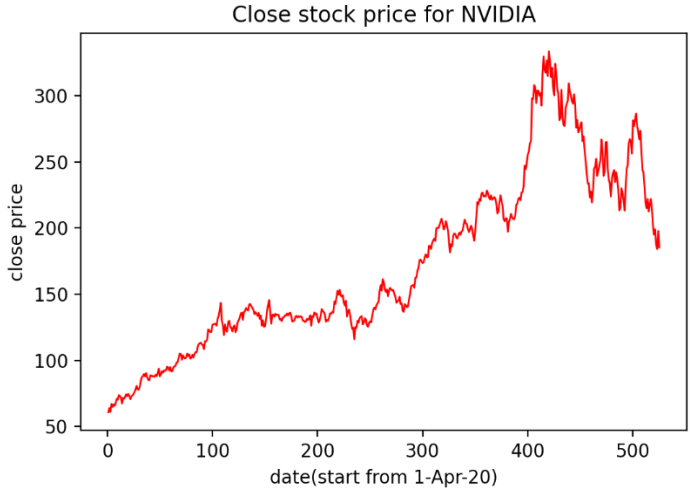

Figure 3. Close stock price of NVIDIA in dollar per day from 1-Apr-2020.

Stockstats from python is used to calculate various indexes for the stock data, including CCI (Commodity Channel Index), KDJ (Stochastic Indicator), and RSI (Relative Strength Index). This requires the dataset to strictly follow the head's form: date, close, open, high, low, and volume [8]. Ten indexes are calculated and stored in the additional rows of the dataset. These indexes would be used as independent variables, and close price would be used as the dependent variable.

2.2 Model

To predict stock price, multiple linear regression is used. It is the model that contains one dependent variable and several different independent variables. The prediction of output value (y) is based on the input values (x), or to say, the features. When a dataset is inputted, the coefficients for each feature can be calculated to establish an equation in multiple linear regression form. The equation of multiple linear regression is shown in Eq. (1). Since the dependent variable can be explained by these independent variables, the output (y) would follow:

$$y = x_1\beta_1 + x_2\beta_2 + \cdots + x_n\beta_n + \varepsilon \tag{1}$$

Where $x, \varepsilon,$ and β are the augmented vectors. And y is the prediction value [9]. To perform a multiple linear regression algorism for stock price, the following assumptions should be established [10]:

There should be a linear relationship between the independent variable and the dependent variable.

It should maintain a constant variance of the errors

The residuals are normally distributed.

Little or no multicollinearity should be performed in the data.

Datasets are divided into two parts. The first part is the train set, which contains the top 80% of the original data, and the second is the test set, which contains 20% of the original data. Based on the proposed model, coefficients for each independent variable (β) would be calculated to set up models used for three stock close prices. Besides, correlation analysis should be used to select significant features, to improve the expression of these multiple linear regression models. In addition, models can avoid being overfitted, and the complexity of the model can be reduced. Finally, the improved models could be used to predict the close price for each stock.

2.3 Metrics

The evaluation of models can be done by using $sklearn.metric$. It can calculate the value of R square, MSE (Mean squared error), MAE (Mean absolute error), and various evaluation parameters. The mathematical descriptions for MSE and MAE are given as follows

$$MSE(y, \hat{y}) = \frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} (y_i - \hat{y}_i)^2 \tag{2}$$

$$MAE(y, \hat{y}) = \frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} |y_i - \hat{y}_i| \tag{3}$$

Here, $\hat{y}_i$ is the predicted value of the $i-th$ sample, and y_i is the corresponding true value. For R square, it can be given as [11]:

$$R^2(y, \hat{y}) = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{4}$$

Where $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$ and $\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \sum_{i=1}^{n} {\epsilon_i}^2$. According to the calculation, the evaluation values for three stocks can be gotten. These values are displayed in Table I.

As shown in Table I, $R^2\ score$ is relatively low for Intel and moderate for AMD and NVIDIA. $R^2\ score$ is the coefficient of determination to evaluate the level of fit of a model [12]. MAE determines the absolute value for the distance from the predicted value to the true value, while MSE indicates the squared value for the distance from the predicted value to the true value. Intel has lower $R^2\ score$, which displayed lower value of MAR and MSE. The situations in AMD and NVIDIA are opposite.

Table I. The evaluation values for each model.

	$R^2\ score$	MAE	MSE
Intel	0.59556	2.7395	11.748
AMD	0.75209	10.4191	160.452
NVIDIA	0.83697	21.613	710.693

3. Results & Discussion

After running the code and building the multiple linear regression model, the predicted value of the close price of each stock can be got, which is $\hat{y}_i$. As a result, the graphs of predicted close price versus date for each stock can be plotted as illustrated in Figs. 4-6.

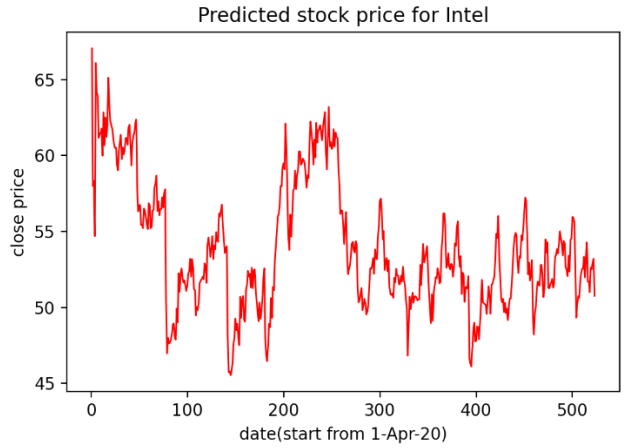

Figure 4. Predicted close stock price of Intel in dollar per day from 1-Apr-2020.

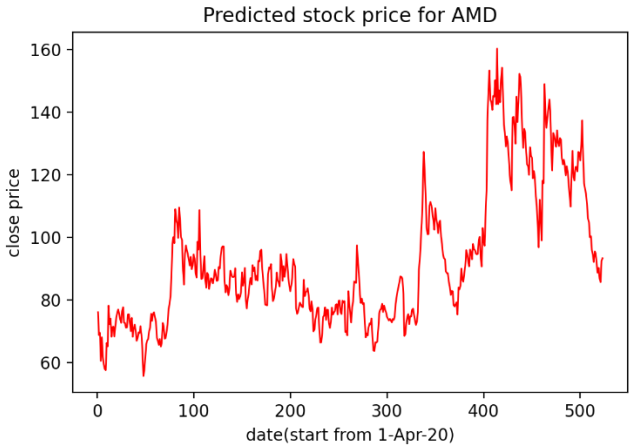

Figure 5. Predicted Close stock price of AMD in dollar per day from 1-Apr-2020.

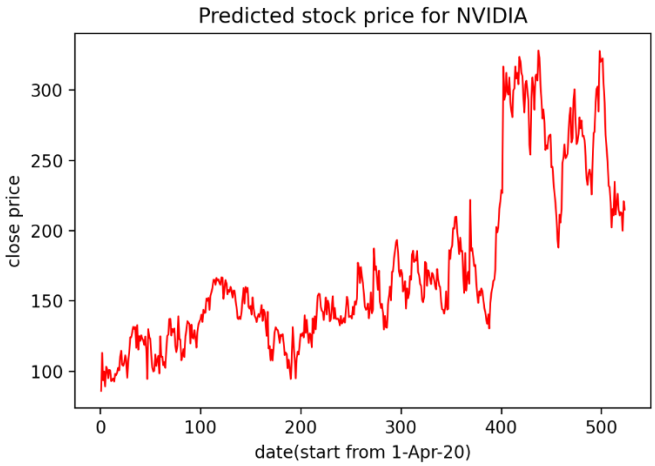

Figure 6. Predicted close stock price of NVIDIA in dollar per day from 1-Apr-2020.

Compared with the true value of the close price, the predicted trend is approximately aligned with the true close price. For some larger fluctuations, it may be failed to predict. In AMD, the close price is in an increasing pattern to reach a peak. Then, the close price falls with fluctuation. When the close price increases, the model cannot to predict the trend. In NVIDIA, this model is the best fit among those three models. It presents the correct pieces with the true ones. It increases and reaches its peak. Subsequently, it begins to decrease, which is more fluctuated, whereas the pattern is apparent in the

predicted one compared with the true one. In Intel, half of the model is comparable with the original one. However, when the close price reaches the peak and begins to decrease, the pattern is not a clear decreasing pattern in the predicted model.

The calculation of the correlation coefficient is a necessary step. The correlation coefficient is used to determine how the independent variable and dependent variable relate [13]. When the results of the correlation coefficient are obtained, some independent variables with an extremely low value of the correlation coefficient would be deleted. There are 8 variables, 2 variables, and 5 variables deleted in AMD, Intel, and NVIDIA, respectively. This is an efficient way of increasing the significance of the model.

After determining a series of evaluation values of these models, this process shows that NVIDIA's model and AMD's model present a higher R^2 score. It shows that factors are well fitted in these two models. However, in Intel, the R^2 score is just about 0.59. This is significantly lower than the R^2 score in AMD and NVIDIA. The model for Intel is not under-fitted. Nevertheless, as shown in Table I, the MSE and MAE are relatively lower in Intel's model and higher in AMD's and NVIDIA's model. This basis implies that the value between the predicted value and the true value or the residual value is greater in AMD's and NVIDIA's model. Greater residual value always implies that this model is not well fitted. The R^2 score is unexpectedly high in these two models. In Intel's model, these residual values are lower, but the R^2 score is low.

The potential explanation is that the calculation of the R^2 score may not include the residual values, which means the R^2 score does not take the residual value into consideration. Therefore, lower residual values could not show that it could have a higher R^2 score. Similar to NVIDIA, it might have a greater R^2 score, but the residual values might be higher among data. As displayed in the graphs, the fluctuations in the predicted functions are much greater and more than the true close price. The result behind this is that the residuals would be large. Although the model displays a relatively similar pattern with the true close price, resulting in the R^2 score in a higher value, the fluctuations make the differences between the predicted value and the true value greater, leading to the high value of MSEs and MAE. The situation is also similar for AMD.

In addition, after the coefficient analysis, several insignificant factors are deleted to improve the model. However, in both AMD's model and NVIDIA's model, the R^2 score is found to decrease after some insignificant factors are deleted. The possible reason behind this is that the current models are overfitted. Therefore, these models perform a higher R^2 score. It should be noted that when several insignificant factors are deleted, the models are improved and become less overfitted. As a result, it is assumed that the R^2 score is decreased for each model. In Intel's model, R^2 score increases as several insignificant factors are deleted. This is expected.

To reduce the complexity of the model and save time, for this experiment, the sample size is 525 days close price for each data. However, if the accuracy is wanted to be better improved, the sample size could be enlarged. This dataset is from 1-Apr-2020 to 1-Apr-2022. The enlarged dataset could be from 1-Apr-2018 to 1-Apr-2022, which may contain 1000 rows of data. On this basis, the significance of the model would be increased. Besides, multicollinearity is a noticeable thing when building a multifactorial model. The appearance of multicollinearity implies that some independent variables are relevant. It will lead to the unstable feature of the model and cause the result of the model to be less accurate and misleading [14]. Therefore, multicollinearity should be considered, and indexes should be calculated to evaluate it. In addition, the model could be further improved by deleting more insignificant factors to reduce the complexity and avoid overfitting. In this case, it increases the significance of the model. Still, a not overfitted model can perform better when this model is used as a prediction tool.

4. Conclusion

In conclusion, this paper mainly focuses on the prediction of stocks using a multiple linear regression model. Three stocks are chosen to use their past close price data to perform a model. Stock

indexes are calculated as independent variables, and models are evaluated by R^2 score, MAE, and MSE. Three models all performed well; one is moderate, and the other two are well fitted. However, these models contain a relatively smaller sample size. The result would be better performed if the sample size was enlarged. In addition, this model can be used to predict other stocks' close prices in the future, and other different types of models could be built to compare with this model in their own results. Therefore, it is still a meaningful model to be made to predict the daily close price. Overall, these results offer a guideline for multi-factorial stock price prediction.

References

[1] A. Ariyo, et al. "Stock Price Prediction Using the ARIMA Model." 2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation, Mar. 2014.

[2] M. Sharaf, et al. "StockPred: A Framework for Stock Price Prediction." Multimedia Tools and Applications vol. 80.12, 2021, pp. 17923-17954.

[3] S. Selvin, et al. "Stock Price Prediction Using LSTM, RNN and CNN-Sliding Window Model." 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Sept. 2017.

[4] S. Vazirani, et al. "Analysis of Various Machine Learning Algorithm and Hybrid Model for Stock Market Prediction Using Python." 2020 International Conference on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE), 9 Oct. 2020.

[5] R. M. Islam, and N. Nguye. "Comparison of Financial Models for Stock Price Prediction." Journal of Risk and Financial Management, vol. 13, no. 8, 14 Aug. 2020, p. 181.

[6] M. Vijh, et al. "Stock Closing Price Prediction Using Machine Learning Techniques." Procedia Computer Science, vol. 167, 2020, pp. 599–606.

[7] S. I. Ao, and International Association of Engineers. International MultiConference of Engineers and Computer Scientists: IMECS 2009: 18-20 March, 2009, Regal Kowloon Hotel, Kowloon, Hong Kong. Hong Kong, Newswood Ltd, 2009.

[8] C. Zhuang. "Stockstats: DataFrame with Inline Stock Statistics Support." PyPI, pypi.org/project/stockstats/. Accessed 14 May 2022.

[9] "ML|Multiple Linear Regression Using Python." GeeksforGeeks, 18 Jan. 2019, www.geeksforgeeks.org/ml-multiple-linear-regression-using-python/?ref=gcse.

[10] D. Deb. "Multiple Linear Regression with Python." Dibyendu Deb, 4 May 2020, dibyendudeb.com/multiple-linear-regression/#Assumptions_for_multiple_linear_regression.

[11] "3.3. Metrics and Scoring: Quantifying the Quality of Predictions-Scikit-Learn 0.22 Documentation." Scikit-Learn.org, 2013, scikit-learn.org/stable/modules/model_evaluation.html#regression-metrics.

[12] A. Akossou, R. Palm, "Impact of data structure on the estimators R-square and adjusted R-square in linear regression." International Journal of Mathematics and Computation vol. 20, 201, pp. 84-93.

[13] P. Bhandari. "A Guide to Correlation Coefficients." Scribbr, 2 Aug. 2021, www.scribbr.com/statistics/correlation-coefficient/.

[14] A. Hayes. "Multicollinearity." Investopedia, 2019, www.investopedia.com/terms/m/multicollinearity.asp.