Stock Price Predictions Using Machine Learning Models

Zebin Guo(✉)

Department of Computer Science, The University of Hong Kong, Hong Kong, China
u3577091@connect.hku.hk

Abstract. Stock predicting is one of the most common topics nowadays. For decades, the stock has been one of the most significant problems in people's life. Many people wish to obtain a higher opportunity to profit by pouring their fortune into the economic market. Consequently, the stock price prediction grows up to be a debated topic. To further investigate the effectiveness of different stock models, this research implements linear regression, decision tree, neural network and LSTM (Long short-term memory) on Netflix stock data. As MSE (mean-squared error) performs well in the regression problem, it is used in this research as a tool for evaluating these models' performances and optimizations. For the specific pattern of the Netflix's stock data (from 2002–2021) the author uses to train the models, the neural network performs the best and gets almost ten times better than the other three.

Keywords: Machine Learning · Stock Price · Predict · Time-Series Data · Neural Network

1 Introduction

The stock price has long been well known for its uncertainty and unstable fluctuations. According to a famous estimate, almost 90% of people lose their fortune in stock markets, including freshmen and seasonal buyers [8]. As a result, the demand for a relatively reliable strategy to predict the stock increases day by day. As predicting price constitutes one of the top fields currently, many researchers have implemented various training figures. In contrast, most of them are concentrating on one model implementation on a single data. For example, one article titled "Stock price forecasting based on LSTM network" written by Huang yucheng and Fang Weiwei, focuses solely on the LSTM model and prompts the impact of the length of time series on the training results. While the content is explicit and worth learning, some improvement points can be made by introducing different models and processing methods, linking the data trend with the absolute error. Instantly, the author decides to move forward for a better insight into the impressions to stock price prediction, close price in particular, from tree regression, linear regression, neural network and LSTM models, and evaluate their performance as well as analyzing the potential causes. This research guided a path to figure out a more suitable model and data selection for stock prediction with machine learning models.

N. Radojević et al. (Eds.): ICAID 2022, AHIS 7, pp. 290–300, 2023.
https://doi.org/10.2991/978-94-6463-010-7_30

2 Background Information

Researchers have been devoting themselves to finding the closest estimation through multiple approaches, consisting of fundamental analysis, technical analysis and machine learning methods.

The fundamental analysis mainly connects with the underlying companies themselves. Through the evaluation of company performance in multiple aspects, it tries to associate the stock price of a company with its market value to place the relative position in the entire market. One of the standard indicators is known as "Buffet indicator" [7], which calculates the "overall market capitalization-to-GDP ratio". It is usually considered the prediction through economic concepts and models, which is preferable by economic-related experts. On the other hand, the technical analysis does not involve any norm of the company's fundamentals. This method achieves the purpose solely based on processing past prices by applying some complex math models, together with skills in the economy, sociology and phycology to evaluate the stock prices.

Unlike the previously mentioned prediction pathways that rely upon high human intelligence to figure out the potential hidden factors influencing the stock price, machine learning depends on digital computations to fit the trend of price movement in the time series. For machine learning, two prime algorithms are artificial neural networks (ANNs) and Genetic Algorithms (GA). Machine learning allows computers to optimize the fitting curve by repeated attempts to minimize the cost function, and it liberates the human brain from tedious mathematical work. However, people are still striving to optimize the fitting model to obtain the matching position for the training data, and reduce or even prevent the possibility of underfitting or overfitting problems [5]. To further investigate the machine learning model's performance on stock prediction, this article mainly focuses on comparing different common machine learning and network models, with optimizations and pre-processing of data.

3 Method

3.1 Data Processing

The Netflix Stock Price (All time) from Kaggle (URL: https://www.kaggle.com/akpmpr/updated-netflix-stock-price-all-time) is used for this research. The filename of this dataset is "netflix.csv". Figure 1 and 2 shows the description of the dataset.

This open-high-low-close chart (also OHLC) [7] serves for Netflix stock price record with a total size of 4881 rows * 7 columns. As shown in Fig. 1, for 7 columns, the first column saves the date of the stock price in date format, while the following six are in decimal format, recording the high price, low price, open price, close price, volume price and adj close price of each day included in the data, respectively. Revealed in Fig. 2, no mismatched or missing data is found in the dataset, so related disposals are superfluous. However, before partitioning training and testing data, data dropping and feature scaling are implemented (Fig. 3).

Date	# High	# Low	# Open	# Close	# Volume	# Adj Close
Self Explanatory	Self Explanatory	Self Explanatory	The open is the starting period of trading on a securities exchange or organized over-the-counter market.	Closing price generally refers to the last price at which a stock trades during a regular trading session.	Volume can be an indicator of market strength, as rising markets on increasing volume are typically.	The adjusted closing price amends a stock's closing price to reflect that stock's value after accounting for any.
23May02 11Oct21	0.41 647	0.35 631	0.38 642	0.37 639	286k 323m	0.37 639
2002-05-23	1.2428569793701172	1.1457140445709229	1.1564290523529053	1.1964290142059326	104790000.0	1.1964290142059326
2002-05-24	1.225000023841858	1.1971429586410522	1.214285969734192	102100000381469727	11104800.0	1.2100000381469727
2002-05-28	1.2321430444717407	1.157142996788025	1.2135709524154663	1.157142996788025	6609400.0	1.157142996788025

Fig. 1. Dataset Columns.

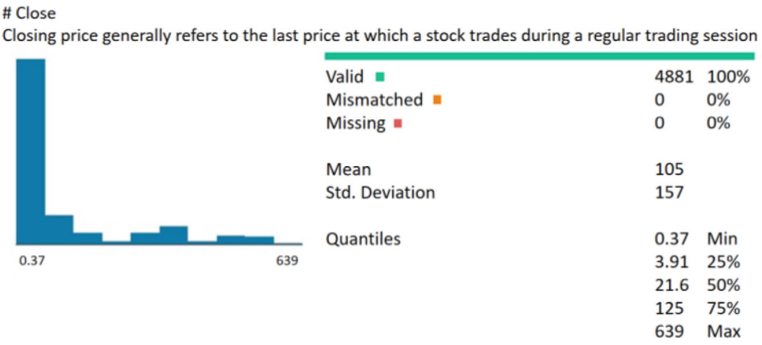

Fig. 2. Close price Data Descriptions.

	Date	High	Low	Open	Close	Volume	Adj Close
0	2002-05-23	1.242857	1.145714	1.156429	1.196429	104790000.0	1.196429
1	2002-05-24	1.225000	1.197143	1.214286	1.210000	11104800.0	1.210000
2	2002-05-28	1.232143	1.157143	1.213571	1.157143	6609400.0	1.157143
3	2002-05-29	1.164286	1.085714	1.164286	1.103571	6757800.0	1.103571
4	2002-05-30	1.107857	1.071429	1.107857	1.071429	10154200.0	1.071429
5	2002-05-31	1.078571	1.071429	1.078571	1.076429	8464400.0	1.076429
6	2002-06-03	1.149286	1.076429	1.080000	1.128571	3151400.0	1.128571
7	2002-06-04	1.140000	1.110714	1.135714	1.117857	3105200.0	1.117857
8	2002-06-05	1.159286	1.107143	1.110714	1.147143	1531600.0	1.147143
9	2002-06-06	1.232143	1.148571	1.150000	1.182143	2305800.0	1.182143

Fig. 3. Head of the dataset.

3.1.1 Data Dropping

Data dropping means throwing out useless data before fitting. The target price is clearly in the decimal presentation in the Netflix stock price dataset. But the data in the 'Date' column is in 'date' format, which shall give no contribution to the price prediction. Also, the 'Adj Price' should have great value for reference, yet in this dataset, the value in the 'Adj Price' column in each row has the same value as one in the "Close" column, so this column is dropped on account of avoidance of data duplication [5].

3.1.2 Feature Scaling

As for the remaining 5 columns, feature scaling is applied to every datum. Min-max scaling and standard scaling are two common scaling approaches. Specifically, the min-max scaling strategy adopted in this experiment instead of standard scaling hinges on the fact that any column in this dataset does not fit the normal distribution thus does not suit standard scaling [2].

Min-max scaling formula:

$$x_{sc} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{1}$$

Where x_{max} and x_{min} corresponds to the maximum value and minimum value of data in a column.

3.2 Regression Model

In this experiment, four different machine learning models are implemented to evaluate the given data and forecast the stock price based on training. Four models are decision tree, linear regression, neural network (NN) and long short-term memory (LSTM).

3.2.1 Decision Tree Model

The decision tree model is one prevalent prediction method labeled with great intelligibility and simplicity. As this experiment tries to predict concrete stock value, the regression tree is applied to determine the value used for tree nodes in separate layers to split out the given data.

3.2.2 Linear Regression Model

Linear regression is a linear method to estimate outputs in scalar form by one/multiple explanatory inputs [9]. This relation is formed by linear predictor functions, while all the parameters are adjusted and learned in data.

Linear regression formula:

$$\mathrm{Y_i} = \beta_0 + \beta_1 \mathrm{X_i} \tag{2}$$

The value of β_0 and β_1 are initialized randomly at the beginning. Through giving data, the model changes the value of β_0 and β_1 by evaluating the fitness of the X-Y line

to the data curve. Specially, alltheparametersandvariablesintheformula can be vectors. If the vector variable is represented in 1 * n size, then n represents the number of input features. In the experiment, n should equal 4 as the input features are columns of "high", "low", "open" and "volume".

3.2.3 Neural Network Model

The neural network implements the third model. In the modern sense, a neural network, named artificial neural network, is composed of neurons and nodes [8]. The artificial neural network, trained by computer, tries to simulate some properties of the biological neural network, which is common in the brain of bio creatures, through forwarding propagation algorithm and backward error transmitting method to train every parameter in multiple layers [3].

3.2.4 Long Short-Term Memory Model

Long short-term memory (LSTM) is a kind of artificial recurrent neural network (RNN). A unit of LSTM has a cell, an input gate, an output gate and a forget gate. Unlike the pure recurrent neural network, the design of the forget gate improves the performance of LSTM on long time-series data, as it uses sigmoid function and activates vectors to control the memory of previous information [4, 6].

The visualization of the LSTM unit is shown in Fig. 4, and the mathematical formula is shown below [11].

$$i_t = \sigma\left(w_i\left[h_{(t-1)}, x_t\right] + b_i\right) \tag{3}$$

$$f_t = \sigma\left(w_f\left[h_{(t-1)}, x_t\right] + b_f\right) \tag{4}$$

$$o_t = \sigma\left(w_o\left[h_{(t-1)}, x_t\right] + b_o\right) \tag{5}$$

As a kind of RNN, LSTM looks at previous sequential data, which is suitable for long time-series prediction. As our dataset has 4881 rows of time, the LSTM method is incorporated in this experiment.

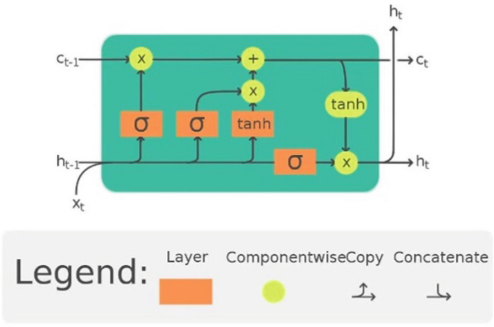

Fig. 4. LSTM Visualization.

3.3 MSE

$$\mathrm{MSE} = \sum_{i=1}^{n} \left(y_i - y_i^p\right)^2 \tag{6}$$

In this research, the best performing model is selected by comparing the MSE value of each model. The smaller the value is, the better the model will be.

4 Results

Figure 5 shows the tangible presence of the target. The training and testing data is split in the portion of 8:2. The number of the testing data is 976, so in the comparisons of data prediction, the last 1000 days are highlighted.

Before getting into predictions using multiple input features, the author employed close price itself as the only input feature to predict the future close price by shifting 25 days back to expect 25 future days. As shown in Fig. 6 and Fig. 7, Decision Tree Regressor and Linear Regressor are used to train. According to MSE estimation, the tree structure predicts the future price with an MSE of 1218.989 (3 significant numbers), while the linear regression model fulfills the prediction with an MSE of 6895.547 (3 significant digits). Consider that for models with multiple inputs, feature scaling is mandatory, and

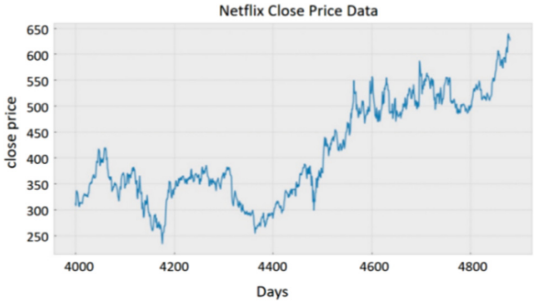

Fig. 5. The original trend of Close Price (starting from date of 2018-04-13, the 4000-th day in the price table).

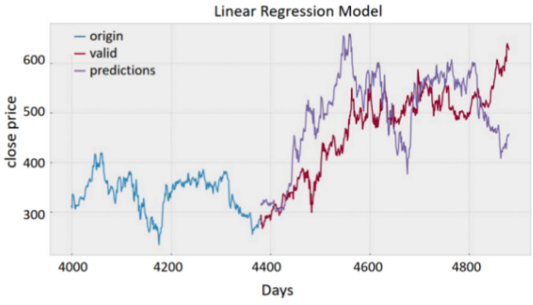

Fig. 6. Linear regression (without feature scaling).

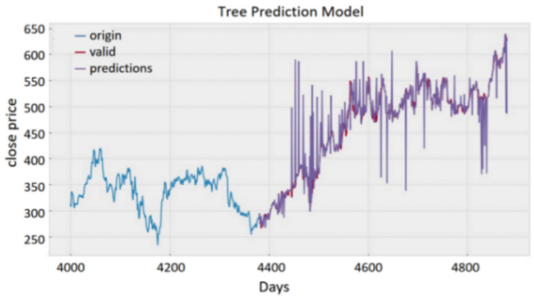

Fig. 7. Decision tree (without feature scaling).

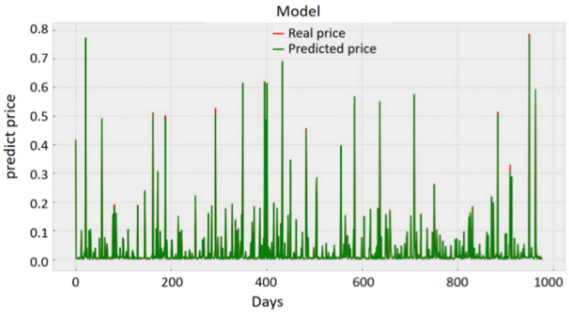

Fig. 8. Tree regression.

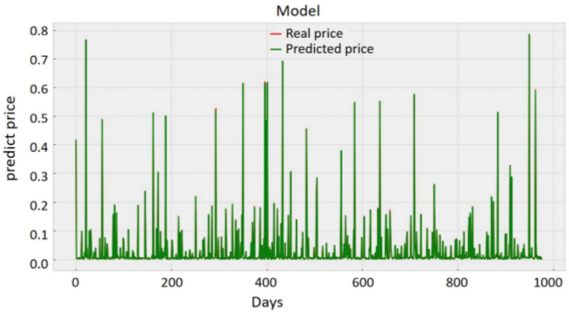

Fig. 9. Linear regression.

these two values are decided by the square of max(df['Close']) − min(df['Close']) which is around 407972, the adjusted MSE for tree regressor is 0.002, while for linear regressor is approximately 0.017 (3 significant numbers).

Factors have to be taken that high accuracy is easily reached for a close-price-to-close-price prediction model, though some shifting work makes the prediction reasonable. For multi-dimension features of the input, Figs. 8, 9, 10 and 11, visualize the performance of the four models.

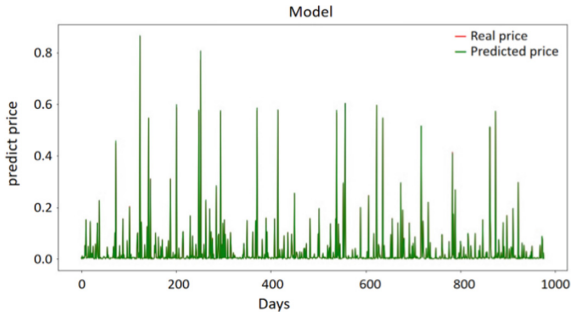

Fig. 10. Neural Network.

Fig. 11. LSTM.

```
#Compute the performance using MSE for tree regressor
real = valid['Close']
pred = valid['predictions']
print('MSE:')
print(mean_squared_error(real, pred))

MSE:
1218.9891234567046
```

Fig. 12. Single feature Tree regression MSE.

Concerning the MSE evaluation, tree regression gets around 2.17 * e^{-5}, 0.89 in decimal format, linear regression gets about 9.10 * e^{-6}, 0.023 in decimal format, NN model gets around 1.45 * e^{-6}, 0.00036 in decimal, and LSTM gets around 0.0022. All numbers presented can be verified in Figs. 12, 13, 14, 15, 16 and 17.

```
#Compute the performance using MSE for linear regressor
real = valid['Close']
pred = valid['predictions']
print('MSE:')
print(mean_squared_error(real, pred))
```

```
MSE:
6895.546649138444
```

Fig. 13. Single feature linear regression MSE.

```
#Compute the performance using MSE for tree regressor
print('MSE:')
print(mean_squared_error(y_test, predictions))
```

```
MSE:
2.1675858839801433e-05
```

Fig. 14. Multi-feature Tree regression MSE.

```
#Compute the performance using MSE for linear regressor
print('MSE:')
print(mean_squared_error(y_test, predictions))
```

```
MSE:
9.085072316607191e-06
```

Fig. 15. Multi-feature linear regression MSE.

```
#find the MSE
print('MSE:')
print(mean_squared_error(y_test, y_predict))
```

```
MSE:
1.4494492695968947e-06
```

Fig. 16. Multi-feature NN MSE.

```
print('MSE:')
print(mean_squared_error(y_test, y_predict))
```

```
MSE:
0.002207990064587865
```

Fig. 17. Multi-feature LSTM MSE.

5 Discussions

A lot of research papers discuss mainly the detailed optimization of the machine learning model, or the implementation of one machine learning model. For a lot of people trying to find regulations and patterns in stock price, too minute research may result out to have limited functions with profound and plentiful fields of knowledge and skills. This experiment focuses mainly on comparing the performance of machine learning models

based on typical stock price data. Presenting some models and approaches of predicting, this article gives some advice on the model and method selection when making standard regression prediction. For example, predicting a specific price by training the same price seems not good even after doing some shifting or employing other adjusting methods. However, there still exists some improvements. This experiment uses different types of price data and volume to predict the close price. Proceeding from the whole situation, all price data types have similar values and trends. As a result, the training result might be lack generality because the model is attempting to collect the condition of other prices and reflect the close price. In this case, a price-to-price prediction may be vulnerable and inadequate when the stock suffers from violating variations in a short time series.

6 Conclusions

In general perception, the stock price data progresses with high linearity. Among the four models used for multi-feature prediction, Neural Network performs the best with the lowest MSE obtainment. For data with high linearity, decision tree and linear regression model with simple modeling procedures also serve well the simplicity of model generation and optimization contributes as the main factor. However, the neglect of issues is not applicable. The terrible performance of LSTM model is far beyond initial expectation due to unavoidable overfitting problems. This experiment does not give out a great strategy to it. During the research, the result obtained from LSTM model gets one sixth value of it was optimized given credit to [1], still the nature of LSTM model lags far behind the other models. So the optimization of LSTM models remains unsearched and not well covered.

Acknowledgements. I'd like to express my deepest thanks to Prof. Vipul Goyal, who introduced me the essential knowledge in the field of machine learning.

References

1. Bashir D, Montañez GD, Sehra S, Segura PS, Lauw J (2020) An information-theoretic perspective on overfitting and underfitting. In: Gallagher M, Moustafa N, Lakshika E (eds) AI 2020, vol 12576. LNCS (LNAI). Springer, Cham, pp 347–358. https://doi.org/10.1007/978-3-030-64984-5_27
2. Geeksforgeeks (August 2021) StandardScaler, MinMaxScaler and RobustScaler techniques – ML. https://www.geeksforgeeks.org/standardscaler-minmaxscaler-and-robustscaler-techniques-ml/
3. Hopfield JJ (1982) Neural networks and physical systems with emergent collective computational abilities. Proc Natl Acad Sci USA 79(8):2554–2558. https://doi.org/10.1073/pnas.79.8.2554
4. Huang Y, Fang W (2021) Stock price forecasting based on LSTM network. Mod Comput 34:51–55+60. https://kns.cnki.net/kcms/detail/detail.aspx?dbcode=CJFD&dbname=CJFDAUTO&filename=XDJS202134008&uniplatform=NZKPT&v=eay5GydBtf5VC4QbSr7pIa4pNyriuaoL8Lj_hxvf-tWJxGAqk03-_dAodrYH8nhE

5. Leung CKS, MacKinnon RK, Wang Y (July 2014) A machine learning approach for stock price prediction. In: Proceedings of the 18th international database engineering & applications symposium, pp 274–277
6. Mehtab S, Sen J, Dutta A (2021) Stock price prediction using machine learning and LSTM-based deep learning models. In: Thampi SM, Piramuthu S, Li K-C, Berretti S, Wozniak M, Singh D (eds) SoMMA 2020, vol 1366. CCIS. Springer, Singapore, pp 88–106. https://doi.org/10.1007/978-981-16-0419-5_8
7. Mislinski J (January 2022) Market Cap to GDP: December Buffett Valuation Indicator. https://www.advisorperspectives.com/dshort/updates/2022/01/13/market-cap-to-gdp-december-buffett-valuation-indicator
8. Rencher AC, Christensen WF (2012) Chapter 10, Multivariate regression – Section 10.1, Introduction. In: Methods of multivariate analysis (Wiley series in probability and statistics), 3rd edn, vol 7090. Wiley, p. 19. ISBN 9781118391679
9. Rockefeller B (February 2014) Technical analysis for dummies, 3rd edn. Wiley Publishing, Inc.
10. Research and Ranking (November 2019) Shocking but true: 90% people lose money in stocks – research & ranking. https://www.researchandranking.com/blog/shocking-but-true-90-people-lose-money-in-stocks
11. Sepp H, Jürgen S (1997) Long short-term memory. Neural Comput 9(8):1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735