Trend-Based K-Nearest Neighbor Algorithm in Stock Price Prediction

Shengtao Gao*

Jiang Nan University, Wuxi, China

*Corresponding author: 2263737271@qq.com

Abstract. This paper introduces a trend-based stock price prediction method that employs the K-nearest neighbors (KNN) algorithm for trend forecasting. Experiments were conducted using a historical stock price dataset, and the prediction performance was evaluated. Experimental evidence suggests that, in relation to accuracy in stock price prediction, the trend-based KNN algorithm exhibits superior performance over conventional machine learning approaches. In addition, the impact of prediction time span on model performance was investigated. The findings suggest that the trend-based KNN algorithm exhibits clear advantages when dealing with predictions over larger time spans.

Keywords: stock price, predict, machine learning, KNN Introduction

1 Introduction

The stock market serves as the heart of the global economy, and its fluctuations profoundly impact the development of the world economy[1]. Therefore, accurate predictions of the stock market, particularly stock price trends, are of significant importance to investors, financial institutions, and even policy-makers[2]. It's widely recognized that the stock market is influenced by numerous factors, such as macroeconomic data, company financial reports, and investor sentiment[3], making its prediction highly challenging.

In recent years , an increasing number of researchers have begun to explore stock price prediction methods based on data science[4]. However, most of these studies focus on using machine learning algorithms to learn from historical data, neglecting the trend information of stock price changes. Therefore, this paper proposes a trend-based stock price prediction method, aiming to improve prediction accuracy by learning and understanding the trends of stock price fluctuations.

C. Chen et al. (eds.), *Proceedings of the 3rd International Conference on Digital Economy and Computer Application (DECA 2023)*, Atlantis Highlights in Computer Sciences 17,
https://doi.org/10.2991/978-94-6463-304-7_78

2 Related work

Stock price prediction has always been a significant research topic within the financial field. Early stock price predictions largely relied on fundamental and technical analysis[5]. However, these methods are mainly dependent on expert experience and subjective judgment, which often limit the accuracy of the predictions.

With the advancement of time series analysis methods, trend forecasting methods have begun to attract the attention of researchers. For instance, in 1989 study, Clemen leveraged time series models such as the Autoregressive Integrated Moving Average (ARIMA) for stock price prediction[6]. In 1992, Brock et al. introduced a stock price forecasting method based on nonlinear dynamic systems, which predicts future trends by analyzing the dynamic behavior of stock prices[7]. Subsequently, in his 2005 study, Tsay proposed a hybrid approach based on both linear and nonlinear models for forecasting prices and trading volumes in the stock market[8]. Nonetheless, these models usually assume that stock price changes follow specific probability distributions, which often do not hold true in practical applications.

In recent years, an increasing number of studies have started to explore stock price prediction methods based on data science. Notably, the K-Nearest Neighbors (KNN) algorithm, also known as a lazy learning supervised technique, has gained prominence. In Altman's 1992 research, the KNN algorithm was successfully applied to stock price prediction, yielding promising results[9]. Following this, some researchers began to explore how to improve the KNN algorithm to enhance the accuracy of stock price prediction. For instance, in their 2001 study, Gavrilov et al. improved the KNN algorithm by proposing a KNN model based on Dynamic Time Warping (DTW), which could better handle the non-linear characteristics of stock price time series, thereby improving prediction accuracy[10]. In 2005, Huang et al. introduced a KNN model based on the Genetic Algorithm that could automatically determine the parameters of KNN, improving prediction accuracy[11]. Liu et al. (2014) [12]proposed a stock price prediction method based on K-nearest neighbor regression and daily/weekly data. They evaluated the performance of the proposed method using different distance metrics and number of neighbors, and compared it with several other popular prediction methods. Singh and Singh (2016)[13] proposed a stock price prediction method based on K-nearest neighbor algorithm and sentiment analysis. They used the sentiment scores of financial news articles as an input feature for the KNN algorithm, and evaluated the performance of the proposed method on real stock market data. Li et al. (2016)[14] proposed a stock price prediction method based on K-nearest neighbor regression and feature selection. They used a correlation-based feature selection method to select the most relevant features for the KNN algorithm, and evaluated the performance of the proposed method on real stock market data.

Building upon these studies, this research will explore a novel approach: the combination of the KNN algorithm and trend forecasting methods, with the aim to enhance the accuracy of stock price prediction.

3 Methodology

3.1 K-Nearest Neighbors Algorithm

The classification rule for K-Nearest Neighbors (KNN) designates a data point x to the category of the nearest data point within a sample set. In contrast, KNN draws upon the majority vote from the K closest points, which in turn impacts the number of neighbors (K) in the classification outcome. KNN discerns the K nearest neighboring points to categorize the group of data that is closest to the test data. Consequently, for KNN, a means to gauge the distance between the query point and the data points in the sample is required. The Euclidean distance, a widely-accepted selection for this purpose, is defined as follows:

$$d(x,y) = \sqrt{\sum_{k=1}^{n} (x_k - y_k)^2} \tag{1}$$

The underlying principle of the K-Nearest Neighbors (KNN) algorithm can be encapsulated as: Given a training set composed of identified data and labels, we input the test data, juxtapose the attributes of the test data against the corresponding attributes in the training set, spot the top K data points in the training set that bear the highest similarity to the test data, and assign the test data to the classification which is most prevalent among these K data points. The algorithm can be articulated as follows:

a) Compute the distance between the test data and every training data point.
b) Arrange them in ascending order based on distance.
c) Pick out the K points that have the shortest distances.
d) Ascertain the prevalence of the categories in which these top K points are situated.
e) Present the class with the utmost prevalence among the top K points as the classification for the test data.

3.2 Trend-Based K-Nearest Neighbors Algorithm

To predict the trend of stock prices, the historical data must first be processed and transformed into a vector form. This transformation is necessary to meet the requirements for predicting stock price trends. Consider x a sequence denoted as $\{m_1(x), m_2(\mathrm{x}) \cdots, m_n(x)\}$, where $m_i(x)$ represents the i-th component of the sequence x. After the historical data undergoes processing and conversion into a vector format, it assumes following structure: $\{m_1(x), m_2(x), \cdots, m_n(x); y\}$, where the initial items denote the m-th distinct attributes of the historical data, and the final item symbolizes the category or objective value of the observed data.

The final price of a stock is presented in a historical detail table format, and the daily ending price of a stock can be perceived as a data sequence. Based on the approach of forecasting relevant entities using time series, we assume that the closing price of any given stock trading day has a relationship with the closing prices of a few previous

trading days. As a result, the past values of the closing price can be depicted as $\{m_1(t), m_2(t), \cdots, m_n(t); y\}$, where the initial m items correspond to the closing prices of the preceding trading days, and the concluding item y denotes the closing price of the respective trading day.

In this way, upon acquiring the details concerning the closing price of trading days, $n-m$ vectors can be generated, thereby resulting in $n-m$ observed data. Subsequently, the KNN algorithm is employed to identify the k data points that are closest to the data of the trading day under prediction from these $n-m$ data., and finally, the average value of the predictive target values of these k data is calculated to obtain the predicted value of the sample.

Hence, for the trading day under prediction, given that the closing prices of the preceding trading days are known, the details of the closing price for the predicted trading day can be procured via this model. The precise computational method is as follows:

a) Compose a time series $\{x_1, x_2 \cdots, x_{n-1}, x_n\}$ with the known closing price data of the n trading days (where x_i represents the closing price of the i-th trading day), and use this set of historical data to predict the closing price x_{n+1} of the $n+1$ trading day. As the closing price of any given trading day invariably has the closest correlation with the closing prices of a handful of preceding trading days, a vector $a_0 = \{x_{n-m+1}, x_{n-m+2}, \cdots, x_{n-1}, x_n\}$ with a length of m is used to predict x_{n+1}, and the k nearest neighbors of $a_0 = \{x_{n-m+1}, x_{n-m+2}, \cdots, x_{n-1}, x_n\}$ need to found.

b) In vector $\{x_1, x_2, \cdots, x_{n-1}, x_n\}$,based on the vector $a_0 = \{x_{n-m+1}, x_{n-m+2}, \cdots, x_{n-1}, x_n\}$, Sequentially take out $n-m$ lengths as subsequences of m : $a_{n-m} = \{x_{n-m}, x_{n-m-1}, x_{n-m+3}, \cdots, x_{n-1}\}, a_{n-m-1} = \{x_{n-m-1}, x_{n-m} \cdots, x_{n-3}, x_{n-2}\}, \cdots, a_2 = \{x_2, \cdots, x_{m+1}\}$, $a_1 = \{x_1, x_2, x_3, \cdots, x_m\}$. Then, the k nearest neighbors of $a_0 = \{x_{n-m+1}, x_{n-m+2}, \cdots, x_{n-1}, x_n\}$ are found in these sub-sequences, and the Euclidean distance is used to measure the proximity of the two vectors.

c) Through calculation, the k nearest neighbors of the vector $a_0 = \{x_{n-m+1}, x_{n-m+2}, \cdots, x_{n-1}, x_n\}$ are found in $a_1, a_2, \cdots a_{n-m}$, denoted as $\beta_1, \beta_2, \cdots, \beta_{k-1}, \beta_k$ .Because $\{x_{n-m+1}, x_{n-m+2}, \cdots, x_{n-1}, x_n\}$ is used to predict x_{n+1}, the element after the last component of these k vectors is considered as a nearest neighbor of x_{n+1}. For example, if $\beta_1 = \{x_1, x_2, \cdots, x_{m-1}, x_m\}$ is a nearest neighbor of a_0, then x_{n+1} is considered as a nearest neighbor of x_{m+1} .Then, according to the KNN algorithm, the k nearest neighbors

$m_1, m_2, \cdots, m_{k-1}, \mathrm{m}_k$ are weighted and averaged to calculate the predicted value of x_{n+1}:

$$x_{n+1} = \frac{\sum_{i=1}^{k} m_i}{k} \tag{2}$$

4 Experience analize

This study selects the daily index data of the Shenzhen Stock Exchange from January 1, 2005 to January 1, 2023, nearly twenty years. The data includes the most important price-volume relationship indicators of the index, such as the opening price (Open), closing price (Close), highest price (High), lowest price (Low), and trading volume (Volume). An example of the data is shown as shown in Table 1:

Table 1. Example of Experimental Data

Date	Open	Close	High	Low	Volume
20221230	13.04	13.16	13.28	12.96	818035.98
20221229	13.07	13.03	13.13	12.85	666890.09
20221228	13.16	13.14	13.38	13.00	791191.98
20221227	12.87	13.11	13.22	12.87	886004.12
20221226	12.99	12.77	13.04	12.71	797119.87

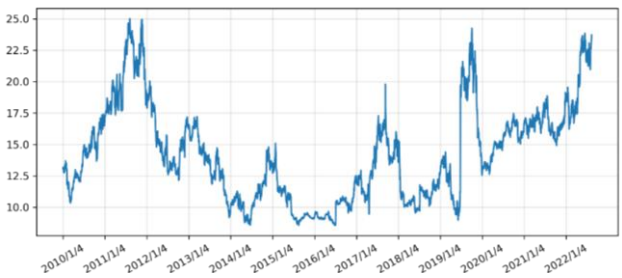

Fig. 1. Stock price chart

In this experiment, the input data is abundant, while the missing data is relatively sparse, accounting for a low proportion of the overall data set. The impact of missing data on the overall data is so small that it can be ignored. Therefore, we choose to directly delete the missing data (Figure 1) .

In order to avoid the influence of dimensions, the experimental data is standardized:

$$x' = \frac{x - \bar{x}}{S} \tag{3}$$

Where, x represents the sample mean, and S represents the sample standard deviation.

The daily data selected for this experiment totals 3088 samples. The dataset is divided into 80% for training (containing 2470 samples) and 20% for testing (containing 618 samples). The training data is used to train the model, and the testing data is used to evaluate the model's performance.

The two most commonly used evaluation metrics are adopted: Root Mean Square Error (RMSE) and Coefficient of Determination (R-Square).

a) RMSE: It is a statistical tool employed to gauge the magnitude of variance between predicted and actual outcomes. It offers a quantifiable way to understand the degree of error present in a predictive model by calculating the square root of the average of squared differences between prediction and actual observation. A smaller RMSE signifies a better fit of the model to the data, indicating the model's predictions are closely aligned with the actual values.

$$RMSE = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(f(x_i) - y_i)^2} \tag{4}$$

b) R-Square: Also recognized as the coefficient of determination, is a statistical index that quantifies the extent to which changes in the dependent variable can be attributed to its relationship with one or more independent variables in a regression model. For instance, an R-Square value of 0.60 signifies that 60% of the variability in the outcome variable has been accounted for by the model, based on its relationship with the input variables.

$$R^2(y,\hat{y}) = 1 - \frac{\sum_{i=0}^{n}\left(y_i - \widehat{y_l}\right)^2}{\sum_{i=0}^{n}\left(y_i - \overline{y}\right)^2} \tag{5}$$

5 Rusult and discussion

The prediction interval for this experiment is 10 days, i.e. the target value to be predicted is chosen as the closing price 10 days after each sample data. The five indicators of stock opening price, closing price, highest price, lowest price and transaction volume are used as feature values input into the model. In order to validate the effectiveness of the proposed method in this chapter, this experiment selects several machine learning algorithms, including: multiple linear regression, SVM algorithm, and KNN algorithm to compare with the algorithm proposed in this chapter. The figure below shows the fitting curves of the prediction results of 100 sample points selected from the test set by various models.

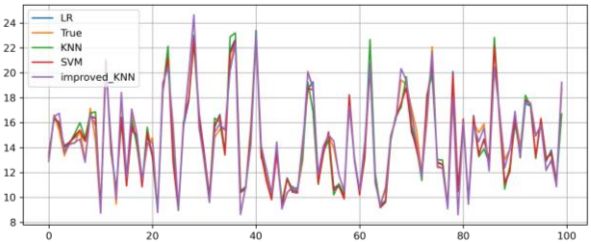

Fig. 2. Fitting results of different algorithms

As shown in Figure 2, by comparing the fitting effect of predicted values and real values through line charts, it can be visually found that the fitting degree of various models are close to the real values, among which the improved KNN algorithm has the best tracking effect. According to the specific evaluation indexes mentioned in the previous section, the following table of comparison of prediction effects of different models is drawn:

Table 2. Example of Experimental Data

	linear regression	SVM	KNN	Improved KNN
RMSE	1.126	1.124	1.237	0.394
R^2	0.903	0.904	0.883	0.988

According to Table 2 due to the relatively long prediction time span and the inherent volatility of stocks, the prediction effects of the above four machine learning model algorithms are not all satisfactory. Among them, the improved KNN algorithm achieves an RMSE of 0.394 and an R-squared value of 0.98, which has obvious advantages over the other machine learning algorithms. The reason is that the improved KNN algorithm takes the historical trend of stock data as information input for model training, which has certain advantages when predicting stock data with a certain time span. Next, this experiment will study the impact of constructing time series of different lengths on the performance of the improved KNN algorithm, and explore the relationship between the algorithm performance and the prediction time span.

In order to study the effectiveness of the algorithm within the prediction time interval, the Pearson correlation coefficient between the closing price on a certain day and the closing price several days later is first calculated, as shown in Figure 3

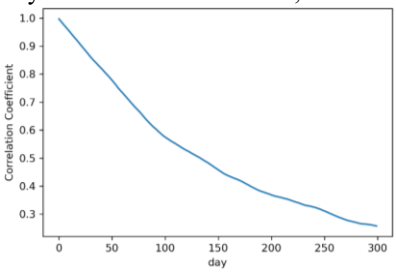

Fig. 3. Correlation Coefficients

In Figure 3, it can be observed that the correlation coefficient between stock prices drops below 0.8 after approximately 50 days and below 0.5 after approximately 100 days. To ensure prediction performance, the forecast horizon of the stock needs to be controlled within 50 days.

Figures 4 and 5 show the variation of the RMSE of the model when the length of the constructed time series and the forecast horizon are changed using the improved KNN algorithm.

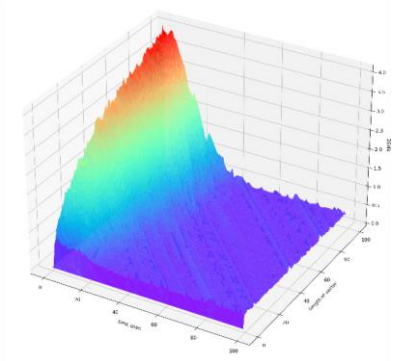

Fig. 4. Variation of RMSE under different parameters

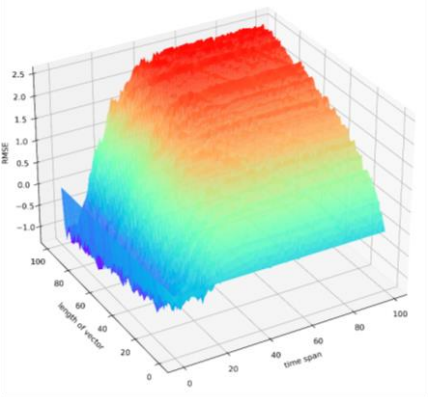

Fig. 5. Variation of RMSE under different parameters

In Figure 4, the x-axis represents the length of the constructed time series, the y-axis represents the forecast horizon, and the z-axis represents the RMSE of the improved KNN algorithm model's predicted results using the experimental data. It can be seen from the graph that as the length of the constructed time series increases, the RMSE of the model decreases, and the accuracy of the model improves. Similarly, as the forecast horizon increases, the RMSE of the model increases, and the accuracy of the model decreases. Additionally, when the length of the constructed time series is 20 or less, and the predicted horizon reaches 60 or more, the model's RMSE exceeds 1.0. However, when the length of the time series reaches 50, within the 100-day predicted horizon

shown in the graph, the RMSE of the model remains stable at less than 1.0. This indicates that the improved KNN algorithm can effectively address the impact of increased time horizons on model predictions through the construction of time series.

In Figure 5, the x-axis represents the length of the constructed time series, the y-axis represents the forecast horizon, and the z-axis represents the difference between the RMSE obtained from the improved KNN algorithm model and the RMSE obtained from the linear regression model. It can be seen from the graph that as the forecast horizon increases, the difference in prediction errors between the two models gradually increases. This shows that as the forecast horizon increases, the advantage of the improved KNN algorithm compared to the linear regression model becomes more pronounced.

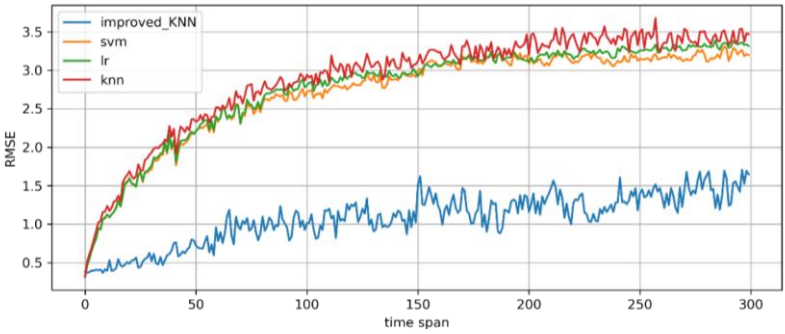

Fig. 6. Changes in RMSE under different prediction time horizons

Figure 6 shows the variation of RMSE of several algorithms under different time horizons. It can be seen that as the predicted time horizon increases, the RMSE of the SVM algorithm, linear regression model, and K-nearest neighbor model gradually increase and stabilize around 3-3.5. In contrast, the RMSE of the improved K-nearest neighbor algorithm increases more slowly and fluctuates between 1-2. This indicates that the improved KNN algorithm has a significant advantage in predicting over larger time horizons.

6 Conclusion

This paper proposes a trend-based stock price prediction method. The method uses the K Nearest Neighbor algorithm to analyze the stock price historical trend and predict stock price changes based on the trend. The main work is as follows:

a) A trend-based stock price prediction method is proposed. The method uses the KNN algorithm to model the trend, and the experiment results show that compared with traditional machine learning algorithms, the prediction accuracy has been improved.

b) The impact of time span on model performance is analyzed. The experimental results show that the correlation coefficient between stock prices drops significantly around 50 days apart, and for long-term predictions with a large time span, the trend-based prediction model has a greater advantage.

Compared with traditional models, the trend-based prediction model proposed in this paper is simpler and more efficient. However, the model proposed in this paper also has some limitations. In building the KNN model, all features are given equal weights, and in the future, the issue of feature weight can be included in the research.

References

1. S. B. Goyal and M. P. Poonia, "Stock market prediction using machine learning algorithms," 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon), Faridabad, India, 2019, pp. 374-378
2. X. Zhang, L. Zhu, H. Liu, and J. Zhang, "Stock Price Prediction Based on Information Entropy with Deep Learning," in IEEE Access, vol. 7, pp. 31711-31724, 2019.
3. Y. Chen, "A Comparative Study of Machine Learning Models for Stock Price Prediction," 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), Chengdu, China, 2019, pp. 1316-1319.
4. T. Fischer and C. Krauss, "Deep learning with long short-term memory networks for financial market predictions," European Journal of Operational Research, vol. 270, no. 2, pp. 654-669, Oct. 2018
5. F. Allen and R. Karjalainen, "Using genetic algorithms to find technical trading rules," Journal of Financial Economics, vol. 51, no. 2, pp. 245-271, 1999.
6. R. T. Clemen, "Combining forecasts: A review and annotated bibliography," International journal of forecasting, vol. 5, no. 4, pp. 559-583, 1989.
7. W. Brock, J. Lakonishok, and B. LeBaron, "Simple technical trading rules and the stochastic properties of stock returns," The Journal of Finance, vol. 47, no. 5, pp. 1731-1764, 1992.
8. R. S. Tsay, "Analysis of financial time series," John Wiley & Sons, 2005.
9. A. H. Altman, "An introduction to kernel and nearest-neighbor nonparametric regression," The American Statistician, vol. 46, no. 3, pp. 175-185, 1992.
10. M. Gavrilov, D. Chetverikov, and V. Melnikov, "KNN and DTW: a new approach to time series clustering," In proceedings of the 5th International Conference on Pattern Recognition and Machine Intelligence, pp. 23-30, 2001.
11. W. Huang, Y. Nakamori, and S. Wang, "Forecasting stock market movement direction with support vector machine," Computers & Operations Research, vol. 32, no. 10, pp. 2513-2522, 2005.
12. Y. Liu, W. Chen, and T. Chen, "Stock price prediction using K-nearest neighbor regression on daily and weekly data," Expert Systems with Applications, vol. 41, no. 1, pp. 147-157, 2014.
13. S. K. Singh and A. K. Singh, "Stock price prediction using K-nearest neighbor algorithm and sentiment analysis," International Journal of Computer Applications, vol. 143, no. 9, pp. 1-6, 2016.
14. Y. Li, K. Wang, and Y. Wang, "Stock price prediction using K-nearest neighbor regression and feature selection," Journal of Intelligent and Fuzzy Systems, vol. 30, no. 5, pp. 3123-3131, 2016.