Stock Price Prediction Using K-NN Clustering and Random Forest Model

黃群翔 R26121081

ABSTRACT

Stock market prediction has long been a significant and challenging issue in the financial field. Due to market uncertainty and price volatility, traditional forecasting methods often struggle to accurately reflect market dynamics. To address these challenges, this study proposes a novel stock price prediction method that combines the k-Nearest Neighbors (k-NN) algorithm with Random Forest and introduces the Average True Range (ATR) as a measure of market volatility. The ATR is used to dynamically adjust the k-value in the k-NN algorithm, allowing for the selection of the most suitable similar samples for prediction based on market conditions. The features of these selected samples are then input into the Random Forest model for stock price forecasting. By comparing this approach with traditional methods, the study demonstrates that the proposed method offers a more flexible and accurate prediction framework. This method significantly enhances stock market forecasting accuracy, providing valuable insights for both future research and practical applications.

1 INTRODUCTION

With the continuous development of global financial markets, stock market prediction has become a key component in financial analysis and investment decision-making. Accurate stock market predictions can help investors seize market opportunities, reduce risks, and enhance returns. However, stock market forecasting faces significant challenges, primarily due to its high uncertainty and complexity. Stock prices are influenced by various factors, including fundamental economic data, market sentiment, political factors, and unexpected events. The interplay of these factors makes it difficult to predict market trends.

Currently, common stock market forecasting methods include statistical models (such as ARIMA) and machine learning approaches (such as support vector machines and decision trees). However, these traditional methods have the following limitations: first, the assumptions of the models are too simplistic. Many statistical models assume that stock market trends follow certain patterns or trends, which contradicts the randomness of the market; second, machine learning methods may overfit when handling nonlinear data and fail to accurately predict unseen data; lastly, many traditional models fail to adequately consider market volatility, which is crucial to the accuracy of stock market predictions.

To address the above challenges, this study proposes an innovative forecasting method that combines the k-Nearest Neighbors (k-NN) algorithm and Random Forest, while introducing the market volatility indicator ATR (Average True Range) to dynamically adjust the parameters of the prediction model. Specifically, the innovation in this study lies in: on the one hand, using the ATR indicator to measure market volatility and dynamically adjusting the k-value in the k-NN model based on the size of ATR; on the other hand, combining Random Forest for prediction to learn the nonlinear relationships between similar samples, further improving prediction accuracy.

The goal of this study is to verify whether the ATR-based and dynamically adjusted k-NN method can more accurately predict the future stock price range and compare its results with traditional forecasting methods, exploring its advantages and limitations.

2 METHODOLOGY

The concept of the model which proposed from Arroyo, González-Rivera, and Maté n.d. can be illustrated through a simple example. As shown in Figure 4, we simulate a generated time series plot, where the red crosses represent the true values of the predicted data points.

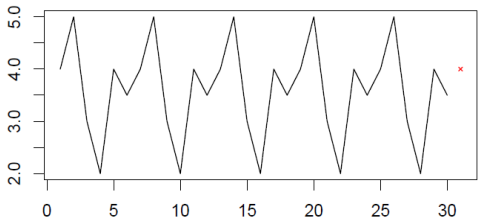

Figure 1: Simulation Time Series Plot

In this example, we set $d = 2$, where the green points represent the reference values and the blue points represent the query values. The main concept of this method is to first identify the k closest data points to the green points. Then, the average or weighted average of the next data values for these k data points is calculated and used as the predicted value. This prediction strategy leverages the trends of similar data to infer the future behavior of the data

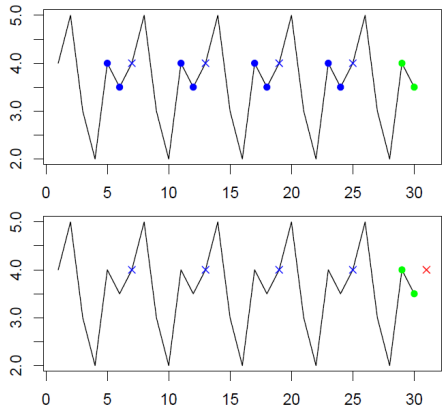

Figure 2: Simulation Time Series Plots with k-NN concept

In this study, we improved existing models by incorporating additional information to enhance the accuracy of stock closing price predictions. Specifically, we adopted the Average True Range (ATR) indicator proposed by J. Welles Wilder Jr., which is primarily used to quantify the volatility of asset prices.

Based on the market's volatility characteristics, we dynamically adjusted the number of similar data points k, thereby improving the precision and adaptability of feature selection. Ultimately, we employed a Random Forest model, using the closing prices of the past three days and the ATR indicator as core predictive features to forecast the next day's closing price.

This approach effectively integrates market volatility with historical price data, aiming to significantly improve the stability and accuracy of stock price predictions.

2.1 Average True Range, ATR

The Average True Range (ATR) is a technical indicator introduced by the renowned technical analysis expert J. Welles Wilder Jr. It is primarily used to quantify the volatility of asset prices. The core concept of ATR is the True Range (TR), which measures the maximum price fluctuation within a single trading day, including the effects of price gaps (gaps caused by market movements)

Steps to Calculate the True Range (TR):.

1. Intraday High-Low Range:

$$Hight - Low$$

Represents the maximum price fluctuation within the day.

2. Absolute Difference Between the Current High and the Previous Close:

$$|Hight_t - Close_{t-1}|$$

Accounts for price gaps caused by opening above or below the previous close.

3. Absolute Difference Between the Current Low and the Previous Close:

$$|Low_t - Close_{t-1}|$$

Similarly, considers the price gap when the opening is below or above the previous close.

4. Maximum Value of True Range (TR):

$$TR = max(Hight - Low, |Hight_t - Close_{t-1}|, |Low_t - Close_{t-1}|)$$

Calculating the Average True Range (ATR):
Using a rolling window approach, the ATR is calculated as the average of the True Range (TR) over the past n days:

$$ATR_t = \frac{1}{n}\sum_{i=0}^{n-1} TR_{t-i}$$

This formula reflects the average level of volatility over the past n days and serves as the basis for the ATR calculation.

2.2 Dynamic Adjustment of k Value

In this study, we dynamically adjust the selection of k in the k-NN model based on the size of the ATR value:

- When the ATR value is high (i.e., the market volatility is large), we choose a larger k value. This increases the sample size, helping to improve the model's stability and reduce the impact of volatility on the prediction results.
- When the ATR value is low (i.e., the market volatility is small), we select a smaller k value to increase the model's sensitivity to local market trends and capture short-term market fluctuations more accurately.

The calculation method for the k value is as follows:

$$k = k_{base}(1 + \frac{ATR}{threshold})$$

The base k value, denoted as k_{base} , is set to 26 by default. The threshold is the volatility reference value, set to 1.5 by default, and it is used to adjust the influence of ATR on the k value. When the ATR value exceeds the set threshold, the k value is adjusted according to the ATR; otherwise, the base value remains unchanged.

2.3 Random Forest Model

Random Forest is a widely used ensemble learning algorithm, belonging to the family of decision trees. It combines the prediction results of multiple decision trees and generates the final prediction through voting or averaging. Random Forest is commonly applied to classification and regression problems and can effectively handle high-dimensional data, nonlinear relationships, and large datasets.

Working Principle of Random Forest

The core idea of Random Forest is to improve model stability and accuracy by generating multiple decision trees and integrating their predictions. The main steps are as follows:

1. **Bootstrap Sampling:**
   Random Forest generates multiple subsets from the original training set using bootstrapped sampling (sampling with replacement). These subsets are used to train different decision trees. Each tree' s training dataset is different, which increases model diversity and reduces the risk of overfitting.

2. **Random Feature Selection:**
   During the construction of each tree, Random Forest randomly selects a subset of features instead of using all the features to decide the splitting nodes. This reduces the correlation between features and enhances the model's generalization ability.

3. **Building Multiple Decision Trees:**
   Random Forest trains one decision tree for each subset. Each tree is trained independently using different data samples and fea-

tures, capturing various patterns in the data and improving prediction accuracy.

4. **Ensemble Prediction:**
   When making predictions on new samples, Random Forest lets each decision tree make a prediction, and then combines the predictions using an ensemble method:
   - In classification problems, it uses a voting method, choosing the class with the most votes as the final result.
   - In regression problems, it averages the predictions of all trees to provide the final forecast.

3 REAL DATA ANALYSIS

3.1 Data Introduction

The data used in this study comes from a BMW stock dataset on the Kaggle platform, covering BMW's daily highest, lowest, and closing prices from 1996 to 2024. The main objective of the study is to estimate the next day's closing price based on the historical stock price data through analysis and modeling.

First, we split the data in chronological order, with 20% of the data selected as the test set to evaluate the performance and accuracy of the predictive model. This splitting method ensures that there is no overlap between the test and training sets, providing a more realistic assessment of the model's predictive ability on unseen data.

3.1.1 k-NN

In the k-NN method, we observed that the daily closing prices, as shown in Figure 3, exhibit a gradual upward trend rather than stable fluctuations.

Figure 3: Close Price Over Time

Therefore, to eliminate the time-dependent changes in stock prices, we applied first-order differencing to the data, as shown in Figure 4. After differencing, the data exhibited a stable trend. Once the data became stationary, we could apply the k-NN method described above to make predictions. In terms of parameter settings, we referred to the study by Arroyo, González-Rivera, and Maté and set the differencing order d to 3, and k to 26.

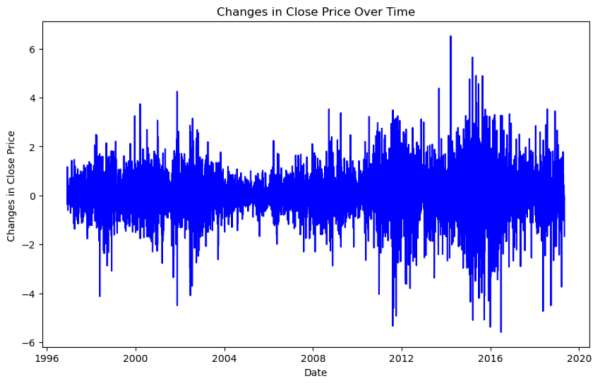

Figure 4: Change in Close Price Over Time

Figure 5 shows the prediction results, with the blue line representing the original closing prices and the yellow line representing the predicted values. As seen in the figure, the predicted values (yellow line) closely align with the actual values (blue line), indicating that the prediction is effective for this dataset.

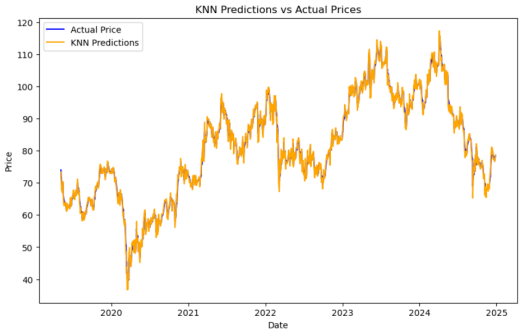

Figure 5: k-NN Predictions v.s. Actual Price

3.2 k-NN + Random Forest Model

In this method, we first calculate the ATR indicator, which reflects the market's volatility. By plotting the time series of ATR values, we observed a gradually increasing trend, as shown in Figure 6.

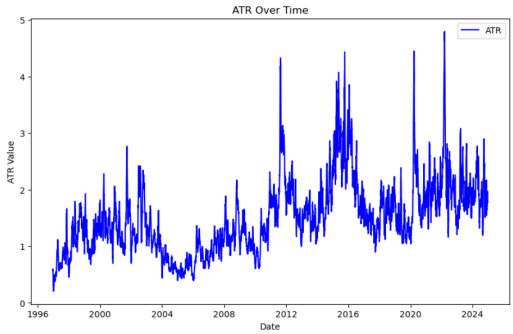

Figure 6: ATR Over Time

To stabilize the data, we applied first-order differencing to the ATR values. After differencing, as shown in Figure 7, the ATR values exhibited a stable trend.

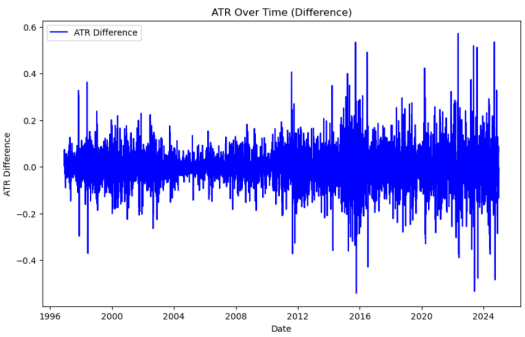

Figure 7: ATR Over Time (Difference)

Next, based on the processed ATR values, we dynamically adjust the k parameter in the k-NN algorithm. When market volatility is high, we select a larger k value and use more similar data points for prediction to capture market trends more accurately. Conversely, when market volatility is low, we choose a smaller k value, reducing the number of required similar data points, and improving sensitivity to the current market conditions. Specifically, the k value is adjusted according to changes in ATR, for example: $(k_1, k_2, k_3, k_4, k_5, \dots) = (49, 49, 51, 53, 52, \dots)$. This setting helps improve prediction accuracy in different volatility scenarios.

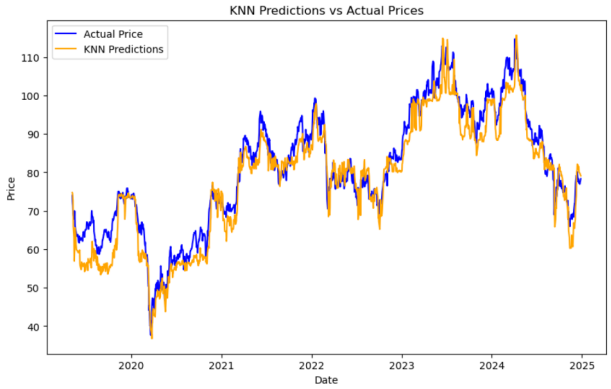

Figure 8: k-NN & Random Forest Model Predictions v.s. Actual Price

After obtaining the aforementioned information, we use the closing prices of the previous three days and the ATR values as features, and input these features into the Random Forest model for training and prediction. Figure 8 shows a comparison between the actual values and the test set predictions, where the blue line represents the true stock closing prices, and the yellow line represents the predicted closing prices from the model. We can observe that the predicted values are very close to the actual values for most of the time, though there are slight discrepancies between the predictions and the actual values at certain moments.

Table 1: RMSE Table

	k-NN	k-NN + Random Forest Model
RMSE	1.66	4.75

Table 1 presents a comparison between the two methods. It can be observed that the RMSE of the k-NN + Random Forest model is higher, indicating that the model's fit is not as good as the original method. A possible reason for this is that the current method for calculating the k-value is not entirely suitable for the forecasting needs of this study, leading to suboptimal prediction accuracy. Addressing this issue will be a key direction for future research and warrants further exploration and improvement.

4 Conclusion

This study integrates the k-NN weighting mechanism with the Random Forest model, proposing a dynamic local prediction framework as an innovative solution for stock price forecasting. By employing an adaptive k-NN approach, the framework dynamically adjusts the neighborhood range based on market volatility (ATR values), enabling flexible selection of local training datasets and improving data utilization efficiency. Meanwhile, the Random Forest model effectively captures the nonlinear relationships and short-term trends in market data, further enhancing prediction accuracy. Although the RMSE increased slightly compared to the original k-NN model, this may be attributed to the dynamic adjustment mechanism.

Future research could explore incorporating additional market indicators and external data, experimenting with alternative weighting strategies or kernel functions, to further improve the model' s adaptability and accuracy. Additionally, applying this framework to other financial and non-financial domains could validate its generalizability and practical value.

This study strikes a balance between flexibility and accuracy, providing a promising solution for addressing market volatility and short-term trend forecasting. It demonstrates significant potential for both research and application across diverse fields.

5 REFERENCE

Arroyo, Javier, Gloria González-Rivera, and Carlos Maté. n.d. ***Forecasting with Interval and Histogram Data. Some Financial Applications.***

Wilder, J. W. (1978). ***New concepts in technical trading systems.*** **Trend Research.**

GitHub link:
`https://github.com/Chun-HsiangHuang/ML_Final_Project`