Scientific Research Publishing

iBusiness, 2022, 14, 75-94
https://www.scirp.org/journal/ib
ISSN Online: 2150-4083
ISSN Print: 2150-4075

Research on Short-Term Stock Price Trends Based on Machine Learning

Lingling Zeng, Yanan Xiao, Shilong Chen

School of Economics, Wuhan University of Technology, Wuhan, China
Email: 1017719953@qq.com

How to cite this paper: Zeng, L. L., Xiao, Y. N., & Chen, S. L. (2022). Research on Short-Term Stock Price Trends Based on Machine Learning. *iBusiness, 14,* 75-94.
https://doi.org/10.4236/ib.2022.142006

Received: April 19, 2022
Accepted: June 3, 2022
Published: June 6, 2022

Abstract

With the increasing emphasis on economic development, China's economy is growing steadily and significantly, and the stock market is becoming more diversified and richer in products and derivatives, resulting in a larger and larger base of investors in the Chinese stock market over the years, with more significant changes in the investment environment of the primary stock market. In the analysis and research around stocks, the fitting and prediction of stock price changes have been one of the keys in the field of stock analysis, however, the current models and solutions for stock price fitting and prediction have not been well received, and are lacking in terms of realistic operability and applicability. In recent years, machine learning and its related models and methods have been widely used in the financial field, which has also promoted the development of stock price fitting forecasting. In order to further improve the accuracy of stock price fitting prediction, this paper introduces the multi-factor prediction model in traditional quantitative stock analysis into the stock price fitting prediction method and improves the general stock price fitting prediction method. This paper finds that the screening of factors that can significantly affect stock prices can indeed be accomplished by using the methodological properties of GBDT and FFM.

Keywords

Machine Learning, Stock Price Fitting Prediction, Quantitative Investment, GBDT, FFW

1. Introduction

In our secondary financial markets, the objectivity of stock analysts' research and analysis is sensitive to their psychological and emotional impact, as well as the additional research consumption and transaction costs due to the increasing

diversification and increase in the size of financial products, which are constantly challenging the traditional methods of financial investment analysis. The use of mathematical methods for quantifiable investment solutions is a more modern financial trading strategy than traditional methods of financial investment analysis. It uses some kinds of independently researched mathematical and statistical methods to abstract the mathematical model for each step of the trading and investment process, and then supplemented with computer hardware and software to enhance the speed and efficiency of the mathematical model calculation. Then finally obtain a more scientific investment solution that is not affected by the analyst's individual bias. These developments mentioned above are also the result of the development of quantitative finance theory and the innovation of computer hardware and software algorithms in the last decade.

Quantitative investment has developed relatively fast thanks to the advancing technology of computers and their derivative computing components. According to statistics, as of the end of the second quarter of 2021, the total asset size of domestic quantitative private equity management was 1034 billion RMB, accounting for 21% of the total size of securities private equity in the same period. At present, 7 of the top 10 global hedge funds are involved in or are quantitative investment institutions. Since 2015, China's options varieties have been gradually enriched and diversified, such as the emergence of securities such as SSE 50ETF options and white sugar options, which have improved the investment coverage of China's options market, complemented the investment strategy space for the majority of financial derivatives investors and reduced the difficulty of complex investment strategies. At the same time, the market effectiveness of China's A-share market is weak and participants have a strong speculative atmosphere, which makes it more convenient for the dissemination and development of quantitative investment theories. Although there is a delayed character in the research of quantitative investment in China compared with foreign capital markets, the field of quantitative investment in China has also been developed rapidly under the continuous opening and maturity of China's capital markets, and its progress space and future are immeasurable.

2. Literature Review

2.1. Predictability of Stock Price

There are two opposing views on the issue of stock price predictability, one believes that stock prices are unpredictable and the other believes that stock prices are predictable. The main theories that hold that stock prices are unpredictable are the random walk theory and the efficient market hypothesis. The efficient market hypothesis was proposed by the American financier Farmer in 1969, which argues that if the stock market reaches efficiency then no investor can make excess returns from stock market trading through technical analysis. Yu (1994) used three different tests in his article to verify the stochasticity of stock price fluctuations in Shanghai and Shenzhen, and the results showed that the

trend of price changes in Chinese stocks is predictable. Wu & Chen (2007) used China's SSE 180 index and all A-share stocks as the research object to build multiple portfolios, and the results of the study found that China's A-share market did not reach weak efficiency. The results found that statistical arbitrage exists in China's A-share market, indicating that China's A-share market has not reached weak efficiency. In addition, Feng (2000), Ye & Cao (2001), Xie et al. (2002), and Lu & Xu (2004) have proved through empirical tests that the Chinese stock market has not reached weak efficiency (price of securities adequately reflects the information implied in a series of historical trading prices and volumes, and it is impossible for an investor to gain excess profit by analyzing past prices.) and investors can predict the future trend of Chinese stock prices through historical stock data.

2.2. Machine Learning Predicting Stock Prices

The trend of stock price movements has long been a key concern for many investors and researchers both at home and abroad, and methods to predict the future trend of stock prices have been continuously explored and researched, some based on statistical methods and others on artificial intelligence and machine learning methods. In general, financial time series data do not necessarily follow some fixed pattern, and therefore numerous statistical methods do not perform well in accurately predicting stock market indices. In contrast to statistical techniques, artificial intelligence methods can handle the stochastic, chaotic, and nonlinear nature of the stock market and are widely used for accurate prediction of stock market indices, such as BP neural networks, random forests, decision trees (DT), genetic algorithms (GA), etc. Zhang & Wu (2009) combined an improved BCO algorithm with BP neural networks to predict stock market movements, and the method performed well in short- and Long-term stock index prediction ability performed excellently. Qiu et al. (2016) applied ANN to Japanese stock market research and used a genetic algorithm and simulated annealing technique, and its prediction accuracy was higher than the traditional ANN. Deng & Wang (2018) used a combined algorithm combining RF and SVM in the study of the U.S. stock market, and experiments showed that such a combined algorithm could improve the prediction accuracy. Shang & Dai (2018) used a combined smoothed ARI-MA-LS-SVM model to forecast the stock prices of four different industries, and the model can effectively predict the short-term price movements. In addition to this scholars such as Hao (2017), Chen (2018) have conducted related studies and the conclusions all prove the feasibility of machine learning methods for stock price forecasting in China.

The above literature all use different machine learning models to predict stock prices. Although the selected data sets, input variables, and predictor variables are not all the same, their findings all show the predictability of stock prices by machine learning methods, and the performance of machine learning methods in predicting stock prices tends to be better compared to statistical techniques,

indicating that machine learning methods can better handle the stochasticity, chaos, and nonlinearity of the stock market and are widely used for accurate prediction of stock market indices.

3. Theories Related to Quantitative Stock Analysis

3.1. Quantitative Stock Analysis Strategies

1) CAPM model

In 1952, Markowitz derived the modern theory of asset allocation by using the variance of returns on assets to judge the risk that a portfolio has, and in this way investigated the way to allocate assets to maximize the expected return of a portfolio in a given risky situation. 20 years later, Sharpe first proposed that stock returns are affected by the fluctuations of market factors, and proposed the CAPM model, in which he studied the relationship between the return on risky assets and other market asset portfolios, which led to the evaluation of the return on risky assets, which is now the common single-factor model in the industry sector.

The CAPM model has a solid mathematical foundation and strict assumptions. According to these conditions provided by the model, it can be inferred that each participant in the capital market has exactly the same risk-return efficient frontier and capital market line. And the systematic risk can be interpreted using the volatility of the integrated market. Based on the above reasoning, the CAPM model can be represented as follows:

$$E\left(r_s\right)=r_f+\beta_s\left(E\left(r_m\right)-r_f\right) \tag{1}$$

$$\beta_s=cover\left(r_s,r_m\right)/var\left(r_m\right) \tag{2}$$

In the model: $E\left(r_s\right)$ represents the expected return of asset portfolio s; r_s, r_m represents the return of risk-free assets and the return of market portfolio M, respectively; β_s represents the sensitivity of risky asset portfolio s to market risk.

2) APT model

The CAPM model provides a single-factor analysis of an asset's return, but in real life, it is clear that many different factors can be found to be influencing asset returns, and a single-factor analysis using the CAPM alone is not sufficient. So Ross proposed the APT model (Arbitrage Pricing Theory model), in which he relaxed the constraints of asset analysis, formally stating that the price of an asset should be influenced by multiple factors. Using this theory, asset returns receive multiple factors in the process of formation, as shown in the following expressions:

$$r_{it}=\alpha_i+b_{i1}F_{1t}+b_{i2}F_{2t}+\cdots+b_{ik}F_{kt}+\varepsilon_{it} \tag{3}$$

In the model: r_{it} denotes the return of security i in period t; F_{kt} denotes the predicted value of factor k in period t; and b_{ik} denotes the magnitude of the sensitivity of security i to factor t.

The above model is further evolved to obtain the multi-factor model.

$$\begin{cases} b_{11}x_1 + b_{12}x_2 + \cdots + b_{1n}x_n = 0 \\ b_{21}x_2 + b_{22}x_2 + \cdots + b_{2n}x_n = 0 \\ \vdots \\ b_{n1}x_1 + b_{n2}x_2 + \cdots + b_{nn}x_n = 0 \end{cases} \tag{4}$$

The newly constructed portfolio has a positive rate of return, recorded in the expression as: $x_1\overline{r_1} + x_2\overline{r_2} + \cdots + x_n\overline{r_n} > 0$.

The above formula illustrates that the factors influencing the return of security f can be decomposed into the joint variation of multiple factors, indicating that the change in the return of a security can be multifactorial.

3) Enamel-Franzi's Five Factor Model

Although the APT model tells us that stock returns can be the result of multiple factors working together, the theory does not show us exactly which factors work together to cause stock price volatility, so investors generally rely on their own stock market investment experience for factor selection and model construction. Until 1992 and 2015, Enamel and Franzi proposed the three-factor and five-factor models successively, by decomposing the unexplained a-terms in the CAPM model, the market value of securities, the book-to-market ratio of securities and market risk were summarized as three factors affecting the fluctuation of securities prices, while Franzi's five-factor model added earnings level risk and investment level risk on the basis of the three-factor model proposed by Enamel that addresses the phenomenon of excess returns of stocks through these two factor characteristics.

Five-factor model:

$$R_i = \alpha_i + b_i R_m + SE(SMB) + h_i E(HMI) + \varepsilon_i \tag{5}$$

Three-factor model:

$$R_i = \alpha_i + b_i R_m + S_i E(SMB) + h_i E(HMI) + rE(RMW) + c_i E(CMA) + \varepsilon_i \tag{6}$$

In model (5) and model (6): R_i is the expected excess return on stock f; R_m is the expected excess return on stock market risk factors; $E(SMB)$ is the expected excess return on a portfolio of small-cap companies over a portfolio of larger-cap companies; $E(HMI)$ is the expected excess return on a portfolio with a high book-to-market ratio over a portfolio with a low book-to-market ratio; $E(RMW)$ is the expected excess return on a portfolio with a high level of profitability over a portfolio with a low level of profitability; $E(CMA)$ is the expected excess return on a portfolio with a low level of investment over a portfolio with a high level of investment; ε_i is the regression residual term.

3.2. Machine Learning Based on Stock Analysis Model

To be able to characterize stock data better, we need multiple dimensions of stock data to characterize stock price changes, so we need to derive the factors that have the most significant impact on stock prices as data dimensions. In var-

ious quantitative stock analysis models, factor screening is often used to obtain stock characteristics. The main idea is to find the class of factors with the strongest correlation with stock returns among a pre-selected set of candidate factor characteristics, and use these factors to construct a stock price analysis model. Stock prices are correlated with company fundamentals, investor sentiment, macroeconomic conditions, market expectations and other influencing factors, and the impact of each factor on stock prices varies.

1) Selection of candidate factors

The candidate stock factor selection process refers to the selection of multiple stock factors that can describe the current stock market environment as candidate factors. Due to the large amount of information in the stock market, there are many factors available for investors to judge, and investors' judgment varies from person to person, and there is a large variance in their expertise in the stock market, which may breed "herding effect" and lead to weak stock market effectiveness. This tells us that the selection of stock price characteristics needs to be done from multiple perspectives.

In this paper, we will use three types of indicators, namely company value indicators, market performance indicators and external environment indicators, as the basis for arriving at the composite factor indicators:

First, Company value indicators. Operating factors reflect the microstructure and production and operation activities of the company to a certain extent, and most of them can be determined by the financial indicators of the company, reflecting profitability, operation, liabilities, cash flow, growth, etc. In order to measure the value of listed companies, this paper will select and calculate current assets, liabilities, accounts receivable, operating income, operating cash flow, earnings per share and other indicators.

Second, Market performance indicators. Market performance factors mainly reflect the process of stock trading. Changes in stock price, trading volume and trading frequency are the sources of information for these factors. They help us to determine the relevant capital flows of stocks, the uncertainties involved and the momentum. At the same time, market performance indicators are also very good at short-term volatility forecasting. Therefore, the risk factor, dynamic P/E ratio, and turnover ratio are selected for calculation in this paper.

Third, External Environment Indicators. External environmental factors are mainly influenced by policies, macroeconomic situations, changes in thinking and technological innovations, etc. These effects are difficult to quantify their impact by scientific means, so they are rarely found in the study. Common external environment factors for stocks include fluctuations in investors' expectations of stock movements, changes in overall economic conditions due to policy natural disasters, etc.

2) Stock Forecasting and Scoring

Multi-factor forecasting models can generally be divided into two types: scoring method and regression method.

First, Scoring method. The most widely used multi-factor stock analysis model is the scoring method. This method scores stocks based on the position of the ranked post-factor, and then gives the weight of each factor to get the overall score of the stock, and the stock with the higher score is used as the desired portfolio. The scoring method is clear and easy to operate and is less affected by outliers, but the disadvantage is that the distribution of the weights of each factor is difficult to grasp and needs to be determined based on the analyst's investment experience.

Second, Regression method. The core of the regression method is to find the magnitude of stock returns affected by different characteristic factors. The multiple regression model is built by regressing multiple factors of stock returns. The model can well quantify the quantitative relationship between factors and returns. The regression coefficients it obtains can be used as the combination weights of each factor. In each adjustment period, the new factor values are substituted into the model to obtain the expectation of each stock price, and the top-ranked stocks are included in the final set of stock picks.

4. GBDT & FFM Fusion Model

4.1. Gradient Boosting Decision Tree

Gradient Boosted Decision Tree (GBDT), also known as Multiple Additive Regression Tree (MART), is a recursive decision tree using iterations. It was first proposed by Jerome Friedman and later improved by him. The algorithm uses the idea of boosting, where n weak learners are constructed at first, and after many iterations, these weak learners can eventually be combined into a strong learner. This strong learner outperforms any of the original weak learners. This algorithm is recognized to have a strong generalization ability.

1) Advantages of the GBDT algorithm

First, it can handle both discrete and continuous values. Second, the advantage of short adjustment time is obvious compared to SVM. Third, the prediction accuracy is relatively high. Finally, the algorithm constructs more loss functions and is highly robust to outliers. In the case of more perturbations and anomalies in the stock market, the GBDT algorithm with these advantages is more suitable. The disadvantage of the GBDT algorithm is that it is difficult to parallelize the linear training process, so to avoid this disadvantage and prevent overfitting, subsampling regularization is required. The GBDT is actually a regression tree, but the classification problem can be viewed as a regression of probabilities, so the GBDT can also be used for classification problem. In summary, the GBDT algorithm will be chosen as one of the stock prediction tools in this paper.

2) Principles of GBDT algorithm

The Carter (CART) regression tree is a weak learner used by GBDT. Each iteration of GBDT is designed to reduce the residuals of the previous model. The iterations can also be trained to build a new model, which is in the direction of

the gradient of residual reduction, as shown in Figure 1.

Given a training data set $=\{(x_1,y_1),(x_2,y_2),\cdots,(x_n,y_n)\}$, fitting is the process of estimating a function $F^*(X)$. We expect that for all $x_i \in X$, there exists $F^*(x_i)$ which make $|F^*(x_i)-y_i|$ tends to 0. In general, the negative gradient of the loss function is often used to fit the loss $|F^*(x_i)-y_i|$.

As can be seen in Figure 1, the GBDT algorithm finds $F^*(x_i)$ by multiple iterations. Let the $m-1^{\text{th}}$ iteration obtain the estimation function $F_{m-1}(x)$ and the loss function $L(y,F_{m-1}(x))$ Then the goal of the mth iteration is to minimize $L(y,F_m(x))=L(y,F_{m-1}(x)+h_m(x))$ by finding the weak learner $h_m(x)$. The condition for the termination of the iteration is that $L(y,F_M(x))$ tends to 0 and the strong learner obtained is $F^*(x)=F_M(x)$.

3) specific calculation steps

The core of the Boosting algorithm is to combine many weak classification models to form a strong classification model. The core idea of GBDT is to combine several simple decision trees into a more complex decision tree. Thus GBDT can be viewed as an additive model consisting of K decision trees:

$$\hat{y}_i=\sum_{k=1}^{k} f_k(X_i) \tag{7}$$

In formula (7), the kth decision tree and F is the function space composed of all decision trees in GBDT. Then the model parameters of the model are:

$$\theta=\{f_1,f_2,\cdots,f_K\} \tag{8}$$

Formula (8) above indicates that the parameters to be learned by GBDT are a set of functions. Define the objective function of the model optimization.

$$Obj=\sum_{i=1}^{N}\ell(y_i,\hat{y}_i)+\sum_{k=1}^{k}\Omega(f_k) \tag{9}$$

In formula (9), $\ell(y_i,\hat{y}_i)$ is the loss function; $\Omega(f_k)$ is the decision tree regularization term, which defines the complexity of the decision tree. To solve this optimization problem for the additive model, a forward stepwise algorithm can be used. The core idea is to use an iterative approach, learning the functional relationships and their structure of only one subtree at each step, and then gradually optimizing the objective function to its optimal point. For additive models,

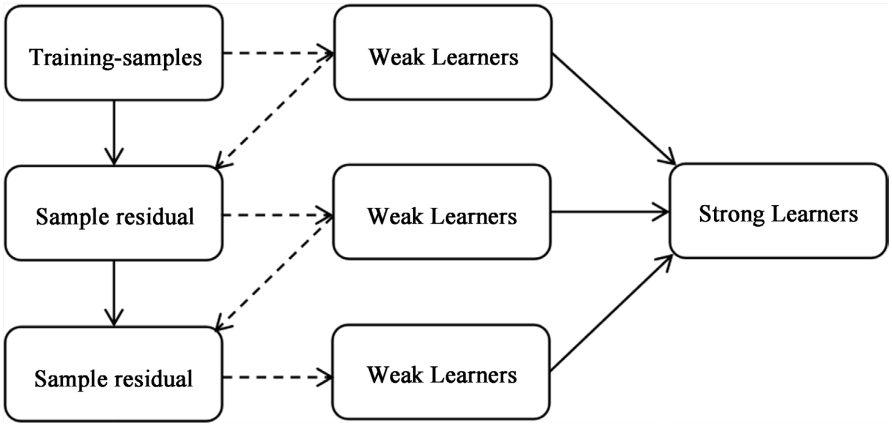

Figure 1. GBDT schematic diagram.

a forward algorithm can be used to solve the optimization problem. The core idea is to use an iterative approach, learning the function relationship and structure of only one subtree at each step, and then gradually optimizing the objective function to the best.

The computational procedure of the forward distribution algorithm is shown below:

Initialization: $\hat{y}_i^0 = 0$; Step 1: $\hat{y}_i^1 = f_1(x_i) = \hat{y}_i^0 + f_1(x_i)$; Step 2: $\hat{y}_i^2 = f_1(x_i) + f_2(x_i) = \hat{y}_i^1 + f_2(x_i)$; …; Step t: $\hat{y}_i^t = \sum_{k=1}^t f_k(x_i) = \hat{y}_i^{t-1} + f_i(x_i)$.

According to the forward stepwise algorithm the objective function can be rewritten in the following form:

$$\begin{aligned} Obj^{(t)} &= \sum_{i=1}^N \ell\left(y_i, \hat{y}_i^{(t)}\right) + \sum_{k=1}^k \Omega(f_k) \\ &= \sum_{i=1}^N \ell\left(y_i, \hat{y}_i^{t-1} + f_t(x_i)\right) + \Omega(f_k) + constant \end{aligned} \tag{10}$$

Further, a second-order Taylor expansion of the function yields:

$$Obj^{(t)} = \sum_{i=1}^N \left[\ell\left(y_i, \hat{y}_i^{t-1}\right) + g_i f_t(x_i) + \frac{1}{2} h_t f_t^2(x_i) \right] + \Omega(f_k) + constant \tag{11}$$

In formula (11), g_i is the order derivative of the loss function $\ell\left(y_i, \hat{y}_i^{t-1}\right)$ with respect to; h_t is the second-order derivative of the loss function $\ell\left(y_i, \hat{y}_i^{t-1}\right)$ with respect to $\hat{y}_i^{(t)}$. Since $\hat{y}_i^{t-1}$ has been obtained from the previous step, $\ell\left(y_i, \hat{y}_i^{t-1}\right)$ in formula (11) is a constant and does not play a role in the function optimization process, formula (11) can be further simplified as:

$$Obj^{(t)} \approx \sum_{i=1}^N \left[g_i f_t(x_i) + \frac{1}{2} h_t f_t^2(x_i) \right] + \Omega(f_k) \tag{12}$$

Formula (12) tells us that, given the loss function $\ell\left(y_i, \hat{y}_i^{t-1}\right)$, there are first-order derivatives g_i and second-order derivatives h_t of the loss function calculated at x_i for all stock samples in the stock data set, which tells us that we can apply specific computational methods to make the loss as close to the minimum as possible in the set of stock samples to obtain the function $f_t(x)$ to be learned at each step, and then the final model can be obtained according to the forward distribution algorithm.

4.2. FM Factorizer

The FFM model is a data analysis and prediction algorithm constructed by people in the process of studying the combinatorial feature problem that exists between the click-through rate, conversion rate, and push system of advertising.

1) The factorization machine

The factorization machine is the base model of FFM and is regularly used in industry. the FM algorithm allows regression and binary prediction. It considers the influence of feature factors and feature factors on each other, and is a nonlinear model. At present, FM algorithm is one of the more satisfactory algorithms in the field of recommendation algorithms, and is widely used in many e-commerce platforms, advertising delivery, and recommendation algorithm services for in-

ternet live streaming.

The advantages of FM algorithm are as follows: FM algorithm can perform reasonable parameter estimation in very sparse data; FM algorithm is compatible, using the extension module it can fit other machine learning algorithms; FM algorithm makes a judgment about the relevance of any feature factor.

2) FM factorizer

In CTR prediction, variables of single heat type are often encountered, which can lead to severe data feature sparsity. The FFM model can solve this problem. The concept of categories, i.e., domains, is introduced in the FFM model. A domain can be considered as a composite of a class of features. In FM, features are independent of each other, while in FFM, it means that the same feature is indexed and the factors generated can all be put into the same domain. For example, the P/N ratio, P/E ratio and P/S ratio in stock features are all financial type factors, and thus these three features can be put into the same domain. In other words, it can be considered that the domain defines multiple feature categories.

For feature x_i, for each of the other factor dimensions of domain f_j, an intrinsic vector $v_i f_j$ is learned. This tells us that intrinsic vectors are associated with both factor dimensions and domains. Assuming that factor dimension x_i interacts with factor dimensions x_j and x_k respectively, the original FM model uses the same intrinsic vector v_i, but the FFM model requires different intrinsic vectors $v_i f_j$ and $v_i f_k$, in which the two models are different.

Let the N factor dimensions of the sample belong to F different domains, then the FFM needs to learn $N^* F$ intrinsic vectors, which is obviously beyond the capability of FM, and thus FM can be considered as a special case of FFM domain $F = 1$. From here we derive its subordinate expressions:

$$y(\vec{x}) = w_0 + \sum_{i=1}^{N} w_i x_i + \sum_{i=1}^{N} \sum_{j=i+1}^{N} \left[v_{i,j}, v_j \right] x_j x_i \tag{13}$$

In turn, the parameters of FFM exist $N^* F^* K$. The FFM model FM has a similar construction concept, but the former is more relevant to the real-world restoration, after all, in the real world, each factor dimension is often intrinsically related, and the introduction of the domain concept allows the FFM model to tap this association to a certain extent, so that the performance of FFM is often better than FM.

4.3. FM Factorizer

1) Secondary generation of features for GBDT models

In essence, when a GBDT model is trained, it actually classifies and combines the sub-trees of the model internally to generate a new combination of them, associating a single factor dimension of the original model with another dimension, and this correlated factor dimension captures more details than a dimension expressed with only one feature, so the new The new feature dimensions generated by the secondary generation of the model have better representational capability.

Let the number of candidate factors be N and the feature factors are indexed

from 1 to *n*. The combination of feature factors in the original stock dataset is $C=\{c_1,c_2,\cdots,c_N\}$, and let the feature vector of the ith input sample in the original stock dataset be X_i. Then we have: $x_i=\{x_{1i},x_{2i},\cdots,x_{Ni}\}$.

GBDT can perform some operations to prevent overfitting during the training process, for example, it rarely uses all the *N* input dimensions completely during the operation, and in most cases GBDT uses each input dimension selectively, which also increases the learning efficiency of the model.

The model maps $x_i\in R^N$ into a *T*-dimensional vector $w_i\in R^T$, also known as feature quadratic generation, which is expressed in mathematical notation as: $\text{GBDT}: x_i\to w_i$. ($w_i=\{w_1,w_2,\cdots,w_T\}$, $w_k\in L_k$).

The original feature input x_i is transformed by the model into feature input w_i, and the element w_k in w_i is the index of the leaf node of the *t*th subtree. We consider w_i as the feature vector after x_i is generated twice. Then the modulus of the new feature vector $\tilde{\omega}_i$ generated by the model twice is equal to the number of child nodes, i.e., the number of new features N' is: $N'=\Sigma_{k=1}^{T}\in L_k$. The feature elements of GBDT are sparse after secondary generation, but the sparsity of the feature elements in this space does not reduce the efficiency and speed of model learning, and this feature also facilitates the feature crossover for subsequent models.

2) Inputs to the FFM model

The FFM model has strict requirements for its inputs, i.e., these vectors need to be composed of domains and their features, as well as the values of these features. We need to make adaptive changes to the new features generated by the GBDT quadratic in order to turn them into structures that can be analyzed using the FFM algorithm.

We then divide the same *T* domains as GBDT in FFM, and randomly select one of them so that its features are those generated quadratically by the corresponding subtree. In another word, we let the sub-tree of GBDT and each domain of FFM construct the correspondence. We index the feature vectors derived from the GBDT model using the solitary heat method, and we can find that it is very reasonable to map the vectors contained in the same tree into the domains of the FFM model, because the elements of its vector expansion have a double mapping relationship with the leaf nodes of the output subtree of GBDT.

Since sparse vectors are large in length and usually inefficient to store because a considerable number of bits are stored with zeros, which is an obstacle for transmission and storage, we use the input format of libSVM, i.e.:

$input=\{(filed:feature:value)_1,(filed:feature:value)_2,\cdots,(filed:feature:value)_T\}$.

The output $w_i=\{w_1,w_2,\cdots,w_T\}$ of the GBDT model is transformed into the above form: $w_i=\{1:w_1:1,2:w_2:2,\cdots,T:w_T:T\}$. Next in this paper, we will use the FFM model to learn and train the new features generated by the GBDT quadratic.

3) Factor selection

In this step, we will use the characteristics of the trained FFM model to select

the optimal combination. For the new features generated quadratically by GBDT, which has T domains and N' features:
$y = \lambda_0 + \sum_{j=1}^{N'} \lambda_1 \tilde{w}_{i,j} + \sum_{j=1}^{N'} \sum_{k=j+1}^{N'} \left(v_{i,k}, v_{k,j}\right) \tilde{w}_{i,j} \tilde{w}_{i,k}$. where $\tilde{w}_{i,j}$ is the jth element of $\tilde{w}_i$. All are Boolean values. The purpose of this chapter is to use the model properties of FFM for fold selection, not to solve the classification and prediction problem, so we only study the higher part of this equation, i.e.: $\sum_{j=1}^{N'} \sum_{k=j+1}^{N'} \left(v_{i,k}, v_{k,j}\right) \tilde{w}_{i,j} \tilde{w}_{i,k}$.

In $\tilde{\lambda}_{jk} = \left(v_{j,k}, v_{k,j}\right)$, the significance of the combination of the characteristic factors is defined as the weight of $\tilde{w}_{i,j} \tilde{w}_{i,k}$, which describes the degree of influence of the change in the combination on the stock price. So we need to pick the group of features $\tilde{w}_{i,j} \tilde{w}_{i,k}$ that satisfy $\tilde{\lambda}_{jk} > \varepsilon$, where ε is the basis of significance for the selection, which means that if a combination of features is to be marked as having a significant impact, then its corresponding weight must exceed the ε value.

5. Experimental Design and Analysis of Results

5.1. Data Selection

Since quantitative modeling of abstract market macro factors is beyond the scope of this paper and there is no very suitable description for the time being, the factors to be selected in this paper are obtained by using company financial indicators and market indicators. In this paper, 600 valid stocks among all listed A-shares in Shanghai and Shenzhen exchanges are selected as the original stock price dataset, and the data are counted by stock trading days as time series data, with dates from November 1, 2021 to April 30, 2022. Next, based on the dataset, various indicators that are helpful for the study of this paper are calculated, specifically, earnings volatility, P/E ratio, P/N ratio, P/S ratio, turnover ratio, market capitalization outstanding, and other 40 candidate factors.

5.2. Data Pre-Processing

In order to simulate the real situation in reality as much as possible and make the model more operable and usable, the above data are appropriately pre-processed in this paper. Since this data set aims to study the short-term trend of stock changes, there is a certain requirement to fit the data noise, so it will be kept as much as possible and only the following processing items are used.

Data addition hysteresis. Since the financial statements of listed companies are not published daily, the access to the corresponding technical indicators will be relatively backward in reality; in fact, in decision-making, we can usually only apply financial information up to the end of the last disclosure cycle for calculation. Therefore, in order to simulate the real stock price forecast, this paper uses the financial report data of the previous disclosure cycle for the estimation of the current period.

Dealing with missing values in the dataset. In this paper, we use a large amount of stock data, and the problem of missing values is inevitable, so we do not try to

complete and interpolate the large number of missing values, instead, we directly eliminate the data samples with a large number of missing values, and this ratio is set at 0.5. For the remaining missing values within a reasonable range, we use the average value of their peers to replace them.

5.3. Data Label Processing

Since both GBDT and FFM are good at dealing with classification problems, there are obstacles to time series analysis of stocks, so at this time we extract the time series as classification labels for classification studies. The classification problem generally requires an input pair shaped as (x_i, y_i) where x_i is the feature vector of stock feature factors, that is, the stock features to be judged; y_i is its corresponding label value. This subsection discusses a feasible label value calculation scheme.

We use the stock price series as the basis for the calculation, so that the time series of stock prices is: $s^{(t)} = \left\{s_1^{(t)}, s_2^{(t)}, \cdots, s_T^{(t)}\right\}$. T is the length of the time series and the series of the fth factor of the i-th stock with respect to time is PI: $x^{(i)[f]} = \left\{x_1^{(i)[f]}, x_2^{(i)[f]}, \cdots, x_T^{(i)[f]}\right\}$ ($i = 1, 2, \cdots, N$). For the tth stock, at each moment t, a sample $x_k = \left\{x_t^{(i)[1]}, x_t^{(i)[2]}, \cdots, x_t^{(i)[N]}\right\}$ can be constructed and its corresponding label can be defined using the increase or decrease of the stock price $\Delta s_t^{(i)}$ in the next W days: $\Delta s^{(t)} = \left| \frac{\frac{1}{w}\sum_{j=0}^{w-1} s_{t+j}^{(i)}}{s^{(i)}} - 1 \right|$. This gives us the tag values and their tag pairs that are available to the model.

5.4. Contrast Analysis

In order to verify whether the factor screening has improved the stock price fitting prediction performance, the stock price fitting prediction accuracy of the unscreened set of factors, the set of factors screened using the original statistical method, the set of factors screened using the single GBDT model method, and the set of factors screened using the GBDT and FFM fusion model are compared here under the random forest prediction model. The stock price fit prediction accuracy of the multi-factor LSTM model was measured using root mean square error.

Using the unscreened set of factors, i.e., direct input of the original 61 candidate

Table 1. Comparison table of factor screening effects.

Factor Screening Method	RMSE of price forecast results
Unscreened factors	1.0343
Factor screening method using statistics	0.8571
Factor screening method using single GBDT	0.8117
Factor screening method using GBDT and FFM fusion model	0.7409

factors to predict stock prices, the final RMSE of the multi-factor LSTM model = 1.0343. Using the original statistical method, i.e., ranking method with covariance coefficients, the 33 factors obtained by screening predict stock prices, and the final RMSE of the random forest model = 0.8571.

Using the single GBDT model, 30 factors were screened to predict stock prices, and the final RMSE of the random forest model was 0.8117. 25 factors were screened to predict stock prices using this paper's GBDT and FFM fusion model, and the final RMSE of the random forest model was 0.7409.

Thus, based on the effect of price prediction, it can be found that the factor screening of the GBDT and FFM fusion model is more effective and has 13.56% in the RMSE of the stock price fitting prediction results compared to the traditional statistical methods. The improvement is 8.7% in the RMSE of the stock price fitting prediction results compared to the single GBDT model.

6. Predictive Models and Analysis of Results

6.1. Predictive Models

1) Random forest model

Random forest models are integrated learning methods. Integrated learning refers to the combination of several individual learners in some integrated way, including subcontracting, boosting, etc. Usually, several "weak learners" are combined to obtain better classification or prediction results. In a random forest model, individual learners are individual decision trees that are integrated in a subcontracting way to obtain a random forest.

The steps are as follows: First, the training set is sampled, and a total of S sub-training sets are drawn, each of which is matched with a decision tree. The random forest model adopts the sub-package method, and the unweighted with the return sampling method, each time a part of the sample is randomly and independently drawn, and then the sample is returned after the drawing. Second, s decision trees are constructed according to S sub-training sets. Similar to the decision tree model, the decision tree is generated according to some feature selection method, but the difference of the random forest model is that the S decision trees are not pruned after generation, and only some features are randomly selected according to some probability principle during the node splitting. Third, when the training data to be classified is input to the random forest model, the classification results of S decision trees are obtained separately, and the classification results with the winning votes are regarded as the output of the random forest model according to the voting method.

The randomness of the random forest model can be fully appreciated from the above construction process, firstly, the sampling process of the training set adopts the sub-packaging method, which can greatly improve the efficiency of the model operation, and secondly, the random feature variables are generated during the construction of individual decision trees, which also simplifies the model design. These two randomization processes enable the random forest model

to solve the problems of overfitting and non-global optimal solutions of individual decision tree models to a certain extent.

2) Decision tree model

The decision tree model is a non-parametric model. The construction of the decision tree model includes the steps of selecting features, generating a decision tree and pruning the decision tree, and using rules to classify and predict the samples to be classified, which has the advantage that the model is not only readable, but also fast in classification.

Feature selection refers to the selection of features that have excellent classification power for the sample data to reduce the depth of the decision tree. It is based on measuring whether the change in purity of the sample dataset after classification is large enough, and common quantifiers of purity change are *Gini* coefficient change. The *Gini* coefficient is defined as: $Gini(D) = 1 - \sum_{i=1}^{r} P_i^2$.

Where D denotes the data set before classification according to a feature, r denotes the number of categories, and the ratio of the number of samples in category i to the total number of samples. The value of *Gini* coefficient is higher when the data in dataset D is more mixed, and conversely, when there is only one category in dataset D, the *Gini* coefficient is equal to the minimum value of 0. The expression for the change of *Gini* coefficient is:

$$\Delta Gini = Gini(D) - \sum_{i=1}^{r} Gini(D_j)\frac{n_{D_j}}{n_D}.$$

Where k denotes the number of feature attributes, D_j denotes the dataset subordinated to the jth feature attribute after classification, and n_D and n_{D_j} denote the number of samples contained in the corresponding dataset, respectively. A larger value of the *Gini* coefficient indicates a better classification of the feature.

Decision tree generation is the process of generating decision tree classification rules according to the established algorithm, and the CART generation algorithm is selected as the algorithm to achieve decision tree generation according to the data situation: Step 1, if the current data set D meets any of the following conditions: all samples have the same class, the sample features are empty, the number of samples is less than a certain threshold, and the depth of the generated decision tree is greater than the set threshold, the leaf node is generated directly and the algorithm is finished; otherwise, it goes to step 2. Step 2, for each feature in the dataset, calculate the *Gini* coefficient variation value for each feature based on the samples. Step 3, select the feature with the largest *Gini* coefficient change as the split node, thus dividing the dataset D into several sub-datasets, and repeat steps 1 to 3 for each sub-dataset.

Decision tree pruning refers to simplifying the decision tree generated in the previous step to reduce the complexity and thus alleviate the overfitting problem.

The CART pruning algorithm is divided into two steps. One is to keep pruning from the bottom of the generated decision tree, and the pruned part is re-

placed by a new leaf node in which the prediction class is determined by the majority class until the root node obtains a sequence of subtrees $\{T_1, T_2, \cdots, T_k\}$; second, a cross-validation test is performed on each subtree to select the optimal subtree as the final decision tree model.

3) Lasso-LR model

The basic idea of the Lasso method is to impose a penalty term on the model coefficients by adding an absolute coefficient function to the model, thus compressing the coefficients with relatively small absolute values to converge to 0. The advantage of the Lasso method is that it not only gives parameter estimates, but also enables variable selection, which greatly improves the efficiency and shortens the computation time. Therefore, the introduction of Lasso method is a practical approach for modeling high-dimensional data. After introducing the Lasso method in the LR model, the expression for estimating the coefficients ω can be written as:

$$\begin{aligned} w &= \arg\min\left\{-l(w) + \lambda\sum_{j=1}^{N}\left|w_j\right|\right\} \\ &= \arg\min\left\{\sum_{i=1}^{m}\log\left(1+\mathrm{e}^{w^{\mathrm{T}}x_i}\right) - y_i w^{\mathrm{T}} x_i + \lambda\sum_{j=1}^{N}\left|w_j\right|\right\} \end{aligned}.$$

A larger λ indicates a higher penalty on the coefficients and a lower number of variables to be included in the model. Ultimately, the problem of estimating the coefficients of the Lasso-LR model is transformed into the optimization problem shown in the above equation, which is usually solved by an iterative algorithm.

4) KNN algorithm

The K-Nearest Neighbor (KNN) algorithm is a classical pattern recognition method and one of the most widely used machine learning methods. KNN algorithm outperforms classification methods such as plain Bayesian classification and support vector machines in dealing with the classification problem of unbalanced data sets, but its performance is still poor.

When a test sample (unknown sample) is given, the pattern space is first searched to find the K training samples (known samples) closest to the test sample, i.e., the K nearest neighbors, and then the selected K nearest neighbors are counted, and if a certain class has the largest number of nearest neighbors, the test sample is judged as that class. Since this computation process is coarse, a more refined statistical method is used, i.e., the sum of the similarity of each class in the K nearest neighbors of the test sample is counted and used as the similarity between this test sample and each class, and finally the test sample is judged to the class with the greatest similarity. In this paper, this finer KNN method is used as a comparison benchmark with the following settings: the dataset contains M classes, each class is denoted as $C_i\left(1 \le i \le M\right)$, and all samples have N attributes.

Step 1, the distance between the test samples and all training samples is calculated. The formula for calculating the distance is: $distance(X,Y) = \sqrt{\sum_{i=1}^{N}\left(x^2 - y^2\right)}$. Where X denotes a test sample, Y denotes a training sample. Step 2, find the K nearest-neighbor training samples with the smallest distance from the test sample. Step 3, calculate the similarity between the K nearest-neighbor training

samples and the test sample respectively. The larger the distance, the smaller the similarity, and vice versa. The formula for calculating the similarity is: $Sim(X,Y)=\frac{1}{1+distance(X,Y)}$. Step 4, the total similarity between each type of nearest neighbor and the test sample is calculated based on the following equation: $Sim(X,C_j)=\sum_{j=1}^{K}Sim(X,Y_j)$ and $\delta(Y_j,C_i)=\begin{cases}1, \text{ if } Y_j\in C_i\\ 0, \text{ if } Y_j\notin C_i\end{cases}$. Step 5, the test sample will be judged as the class with the greatest similarity according to the following equation: $f(x)=\arg\max\left(Sim(X_j,C_i)\right)$.

5) Plain Bayesian algorithm

The plain Bayesian classification model (NBC) is one of the most widely used Bayesian classification models. The model is described as follows: Let the training set contain m classes $C=\{C_1,C_2,\cdots,C_m\}$ and n conditional attributes $X=\{X_1,X_2,\cdots,X_n\}$. Suppose all the conditional attributes X are treated as children of the class variable C. A given sample $\{X_1,X_2,\cdots,X_n\}$ to be classified is assigned to class $C_i(1\le i\le m)$ if and only if: $P(C_i\mid X)>P(C_j\mid X)$ $(1\le i,j\le m;i\ne j)$ holds. According to Bayes' theorem, the posterior probability of class C_i is: $P(C_i\mid X)=\frac{P(C_i)P(X\mid C_i)}{P(X)}$.

If the probabilities of the classes on the training set are not known in advance, each class probability can be assumed to be equal, $P(C_i)=P(C_j)$, and $P(X\mid C_i)$ is maximized. If the probabilities of each class are known, $P(C_i),P(X\mid C_i)$ is maximized. $P(X)$ in above is constant for all categories and can be omitted, so we have: $P(C_i\mid X)=\frac{P(C_i)P(X\mid C_i)}{P(X)}\propto P(C_i)P(X\mid C_i)$.

According to the assumption that the conditional attributes of the plain Bayesian classification algorithm are independent of each other, there are $P(C_i\mid X)\propto P(C_i)\prod_{i=1}^{n}P(X\mid C_i)$, in it, $P(C_i)=N_i/N$, N_i is the number of instances of class C_i in the training samples, N is the total number of training samples. So the NB model is.: $P(C_i\mid X)=\max P(C_i)\prod_{i=1}^{n}P(X\mid C_i)$.

6.2. Analysis of Results

1) Introduction of evaluation indicators

To evaluate the performance of our model, we use three evaluation metrics, namely Accuracy, F-value, and AUC. Before introducing these metrics, we introduce four basic elements that are widely used to evaluate the classification performance of the model: True Positive (TP), False Positive (FP), False Negative (FN), and True Negative (TN). TP refers to the number of samples with positive labels that are also predicted to be positive; FP refers to the number of samples with negative labels that are predicted to be positive; FN refers to the number of samples with positive labels that are predicted to be negative; and TN refers to the number of samples with negative labels that are also predicted to be

negative. TN refers to the number of samples with negative labels that are also predicted as negative labels. Using the above four basic elements, we define the accuracy as: $\text{Accuracy} = (\text{TP}+\text{TN})/(\text{TP}+\text{FP}+\text{FN}+\text{TN})$.

Precision is a measure of the degree of accuracy of the model prediction. The F-value is defined as follows: $\text{F} = 2 * \text{Accuracy} * \text{Recall}/\text{Precision} * \text{Recall}$.

The above $\text{Accuracy} = \text{TP}/(\text{TP}+\text{FP})$, $\text{Recall} = \text{TP}/(\text{TP}+\text{FN})$. The F-value is a weighted average measure of precision and recall, ranging from 0 to 1. The larger the F-value, the better the classification model is.

The full name of AUC is Area under the Curve. Usually, the curve here refers to the subject operating curve. In contrast to accuracy, recall, F-value, and other evaluation metrics that depend on judgment thresholds, AUC does not have this problem. The range of AUC is from 0 to 1, and larger values indicate better classification models.

2) Analysis of results

We measure the results of our model using three metrics as benchmarks respectively.

From Table 2, we can see that according to accuracy, RF model achieved the best average effect of 0.983 and Nb achieved the worst average effect of 0.927.

From Table 3, we can see that RF achieved the best average effect of 0.975, while NB achieved the worst average effect of 0.894.

From Table 4, we can see that RF achieved the best average effect of 0.981, while NB achieved the worst average effect of 0.920.

From the above three tables, we can see that the random forest model has achieved the best effect on the five indicators, and Nb has achieved the worst effect on the five indicators.

Table 2. Accuracy

Number of cycles	RF	NB	DT	LR	KNN
1	0.983051	0.926891	0.98128	0.974197	0.976474
2	0.98381	0.926891	0.981027	0.974703	0.976221
3	0.983051	0.926891	0.981027	0.974197	0.976221
4	0.983304	0.926891	0.98128	0.974197	0.976221
5	0.982545	0.926891	0.980268	0.974197	0.975968

Table 3. F-value

Number of cycles	RF	NB	DT	LR	KNN
1	0.975322	0.894487	0.972774	0.962555	0.965897
2	0.976453	0.894487	0.972396	0.963262	0.965517
3	0.97534	0.894487	0.972376	0.962555	0.965517
4	0.9757	0.894487	0.972794	0.962555	0.965517
5	0.974604	0.894487	0.971302	0.962555	0.965138

Table 4. AUC.

Number of cycles	RF	NB	DT	LR	KNN
1	0.980288	0.920226	0.978589	0.971272	0.974052
2	0.981388	0.920226	0.978222	0.971658	0.973685
3	0.980462	0.920226	0.978048	0.971272	0.973685
4	0.980655	0.920226	0.978763	0.971272	0.973685
5	0.979902	0.920226	0.977469	0.971272	0.973318

In conclusion, by using the methodological properties of GBDTt and FFM, we can indeed complete the screening of factors that can significantly affect the stock price. At the same time, we can use the idea of compound learning algorithm of these two models to build a stock factor screening model using the combined model of gradient lifting decision tree and FM factorization machine, and can effectively screen the factors affecting the stock price, Finally, the subsequent prediction model of this model is best to use the random forest model to obtain the best prediction results.

6.3. Deficiencies

First, there are difficulties in the practical application of traditional methods. Traditional analysis methods for forecasting stock price movement trends rely heavily on the analyst's long-term experience in observing stock market development, which requires a great deal of professionalism, i.e., analytical ability. Second, new types of forecasting models are emerging. With the continuous development of mathematical finance and computer technology, more powerful stock description forecasting models have been generated and therefore stock price fitting forecasting models are gradually absorbing them into the analysis system. Third, the limitations and singularity of factor pool construction. There are many existing major stock-influencing factors, but most studies tend to focus on a single factor. Finally, there is not enough use of deep learning algorithms. This paper finds more studies using traditional machine learning models, while there are fewer cases of using emerging AI deep learning algorithms.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

Chen, W.-H. (2018). Comparative Study on the Effect of Volatility Prediction of SSE Composite Index Based on Deep Learning. *Statistics and Information Forum, 33,* 99-106.

Deng, F. X., & Wang, H. L. (2018). Application of LSTM Neural Network in Stock Price Trend Prediction—A Study Based on Individual Stock Data in the US and Hong Kong Stock Markets. *Financial Economics,* 96-98.

Feng, L. C. (2000). The "Intra-Week Effect" in the Chinese Stock Market. *Economic Research,* 9-12

Hao, Z. Y. (2017). Stock Forecasting Method Based on Improved Support Vector Machine. *Journal of Jiangsu University of Science and Technology (Natural Science Edition), 31,* 339-343.

Lu, R., & Xu, L. B. (2004). A Study of the Unbalanced Response of "Bull" and "Bear" Markets to Information. *Economic Research, 39,* 65-72.

Qiu, M., Song, Y., & Akagi, F. (2016). Application of Artificial Neural Network for the Prediction of Stack Market Returns: The Case of the Japanese Stock Market. *Chaos, Solitons & Fractals, 85,* 1-7. https://doi.org/10.1016/j.chaos.2016.01.004

Shang, W. P., & Dai, X. (2018). Research on Stock Price Forecasting Based on Smoothed ARIMA-LS-SVM Combined Model. *Regional Finance Research,* 17-23.

Wu, Z.-X., & Chen, M. (2007). A Statistical Arbitrage Test of the Weak Effectiveness of the Chinese Stock Market Closed. *Systems Engineering: Theory and Practice, 27,* 92-98.

Xie, B. H., Gao, R. X., & Ma, Z. (2002). An Empirical Test of Stock Market Effectiveness in China. *Quantitative Economics and Technical Economics, 19,* 100-103.

Ye, C. H., & Cao, Y. J. (2001). Application of Hurst Index in Stock Market Effectiveness Analysis. *Systems Engineering, 19,* 21-24.

Yu, Q. (1994). Market Efficiency, Cyclical Anomalies and Stock Price Volatility: An Empirical Analysis of the Shanghai and Shenzhen Stock Markets. *Economic Research,* 43-50.

Zhang, Y., & Wu, L. (2009). Stock Market Prediction of S&P 500 via Combination of Improved BCO Approach and BP Neural Network. *Expert Systems with Applications, 36,* 8849-8854. https://doi.org/10.1016/j.eswa.2008.11.028