International Journal of Financial Engineering
Vol. 8, No. 2 (2021) 2150027 (17 pages)

DOI: 10.1142/S2424786321500274

World Scientific
www.worldscientific.com

Predicting the trend of stock index based on feature engineering and CatBoost model

Renzhe Xu*, Yudong Chen†, Tenglong Xiao‡, Jingli Wang§ and Xiong Wang¶
Institute for Advanced Study, Shenzhen University
Shenzhen, Guangdong, P. R. China

Received: 28 December 2020; Accepted: 19 April 2021
Published: 25 May 2021

Abstract

As an important tool to measure the current situation of the whole stock market, the stock index has always been the focus of researchers, especially for its prediction. This paper uses trend types, which are received by clustering price series under multiple time scale, combined with the day-of-the-week effect to construct a categorical feature combination. Based on the historical data of six kinds of Chinese stock indexes, the CatBoost model is used for training and predicting. Experimental results show that the out-of-sample prediction accuracy is 0.55, and the long–short trading strategy can obtain average annualized return of 34.43%, which is a great improvement compared with other classical classification algorithms. Under the rolling back-testing, the model can always obtain stable returns in each period of time from 2012 to 2020. Among them, the SSESC's long–short strategy has the best performance with an annualized return of 40.85% and a sharp ratio of 1.53. Therefore, the trend information on multiple time-scale features based on feature engineering can be learned by the CatBoost model well, which has a guiding effect on predicting stock index trends.

Keywords: Stock market index; CatBoost model; feature engineering; clustering.

1. Introduction

In the field of quantitative investment, the prediction of stock market has always been one of the most interesting and challenging issues for investors and related researchers. However, the efficient market hypothesis restricts the usefulness of

*Corresponding author.
Email addresses: *1900391001@email.szu.edu.cn, †chenyudong2019@email.szu.edu.cn, ‡xiaotenglong@email.szu.edu.cn, §2060391006@email.szu.edu.cn, ¶wangxiong8686@qq.com

stock market prediction to some extent (Malkiel and Fama, 1970). The efficient market hypothesis assumes that stock prices are only correlated with the latest information and rational expectations, and recent disclosures about a company's prospects are reflected almost immediately in current stock prices. Therefore, the change of stock price is due to the release of new information. As for the daily fluctuation of stock price, it can be described by a statistical process called random walk, which means that daily deviations from the center value are unpredictable. In fact, actual market conditions differ from the hypothesis of unpredictability, more and more papers prove that the stock market is predictable. William first used Bootstrap method to re-sample the rate of return model with a specific structure, in order to obtain excess returns of technical trading rules and empirical distribution of other test statistics. This study examined the data of Dow Jones industrial index from 1897 to 1986, and empirically tested two most common technical analysis trading rules namely moving average rule and resistance line support rule which shows that the technical analysis is helpful in predicting stock returns (Brock *et al.*, 1992). Later, scholars have also come to a similar conclusion on the predictability of stock prices, that is, the operation of stock prices has its inherent law, through certain methods such as technical analysis, excess returns can be obtained (Marshall and Cahan, 2005).

Traditional stock market predicting methods can be divided into two broad

Traditional stock market predicting methods can be divided into two broad categories: fundamental analysis and technical analysis. Fundamental analysis refers to the analysis of macroeconomic aspects, the industry in which the company's main business is located, the level of competition in the same industry, the internal management level of the company, and so on. Technical analysis is used to analyze past data of price and volume, then to predict price trends in the future. This type of analysis focuses on the composition of charts and formulas to capture major trends and identifies buying or selling opportunities by estimating market cycles. Most stock predicting models are based on technical analysis methods. A study on the Markov model for forecasting the Shanghai Stock Exchange Index confirmed that its accuracy over a sufficiently long period was higher than that of the general technical analysis method, and good results can be obtained when the model is used to predict the recent price trend of a single stock (Guan and Zhao, 2005). Caginalp and Desantis (2011) quantitatively identified key aspects of technical analysis, such as trends and resistance, which were scientifically effective. Moreover, Bai (2009) applied the ARIMA model to the short-term trend of stock index futures and showed that it can also be efficient and has less prediction error.

Stock returns are characterized by complex nonlinearity and high noise (Ballings *et al.*, 2015). Especially in the Chinese stock market, a large number of individual investors bring additional challenges to obtain reliable prediction. With

the development of information technology, various machine learning techniques for nonlinear data are applied to the prediction of stock market, which can provide another idea to this issue. Huang *et al.* (2005) predicted the Nikkei 225 index at the weekly level using support vector machine (SVM), Linear Discriminant Analysis, Quadratic Discriminant Analysis and BP Neural Networks. The experiment results showed that SVM has a better classification effect than other classifiers. Hassan *et al.* (2007) used artificial neural network (ANN) to convert the daily stock price into an independent numerical set as the input of the hidden Markov model (HMM) and used genetic algorithm (GA) to optimize parameters, so as to obtain a model combining HMM, ANN and GA to predict financial market behavior. Hao and Gao (2020) proposed a new end-to-end hybrid neural network model based on multiple time-scale feature learning to predict the price trend of the stock index. The prediction performance of the hybrid neural network is better than that of the short time scale. Nti *et al.* (2020) made an extensive comparative analysis of the prediction results of various ensemble techniques in the stock market. By using decision tree, SVM, and neural network, 25 different ensemble regressors and classifiers were constructed. They compared the execution time, accuracy, and error measurement of stock data from January 2012 to December 2018 on the Ghana Stock Exchange (GSE), the Johannesburg Stock Exchange (JSE), the Bombay Stock Exchange (BSE-SENSEX), and the New York Stock Exchange (NYSE) from January 2012 to December 2018, and concluded that the ensemble algorithm is of great significance in the field of stock market prediction.

The prediction performance in financial markets depends not only on the machine algorithms or models used but also on the input features. Patel *et al.* (2015) predicted the trend of stock index in the Indian stock market. They compared four prediction models, ANN, SVM, random forest, and naive Bayes, respectively, with two kinds of input features, technical indicators, and trend deterministic data. The experimental results showed that the trend deterministic data has better prediction performance in each model. Machine learning methods mostly use daily closing price data to construct technical factors or fundamental data to construct long-term effective factors. In the short term, especially the trend of stock price forecast on the second day, there are inevitable lagging effects, which are not sensitive to the abrupt change of short-term trend. From the technical point of view, the intraday stock price trend can provide a large number of timely and effective game information of the day's long short trading for the next day's stock trend prediction. Therefore, this study innovatively develops features based on intraday price series categories, which can effectively provide prediction information for the short-term trend of the stock index. Combining with the Day-of-the-Week Effect of the Chinese stock market (Cai *et al.*, 2006), we use the day of the week as an

additional categorical feature, and construct a more microscopic and lower delay categorical feature combination.

For the classification of categorical features, we choose to use Categorical Boosting (CatBoost) model, which is an ensemble learning algorithm based on the gradient boost decision tree. On the one hand, it is an order Boosting algorithm which is obtained based on the improvement of the standard gradient Boosting algorithm and can avoid the target missing. On the other hand, it is a new method to deal with categorical features. The previous gradient boosting algorithm is faced with a special problem of missing target, that is, the prediction model obtained relies on the targets of all the training samples, which leads to the shift of the predicted sample distribution from the training sample distribution, and ultimately the shift of the predicted results from the training model. CatBoost solves this problem and does it better than XGBoost and LightGBM. In addition, CatBoost also has a fast, scalable GPU version, which trains the model with GPU-based gradient enhancement algorithms to reduce over-merging and achieve fast prediction (Prokhorenkova *et al.*, 2018; Dorogush *et al.*, 2018). CatBoost, with its excellent properties, is the primary way to learn about problems with heterogeneous characteristics, noisy data, and complex dependencies, and is widely used across industries. In the field of biomedicine, CatBoost algorithm is widely used in the detection of bioactive substances and medical diagnosis (Postnikov *et al.*, 2020; Al-Sarem *et al.*, 2020). In the field of social science, CatBoost algorithm is widely used for behavior detection and determination (Kang *et al.*, 2019; Dou, 2020; Punmiya and Choe, 2019). In addition, there are scholars who built three ensemble predicting models, namely XgBoost, LightGBMm, and CatBoost, based on the data of eight consecutive years of high transfer stocks in the Chinese stock market. The experimental results show that the three models are far better than the traditional logistic model, and the fusion model obtained from the three models is more effective (Ni *et al.*, 2020).

The rest of this paper is structured as follows. In Sec. 2, we give the judgment method of interday minute-level price series category and construct feature combination. In Sec. 3, we introduce prediction models used in this research. In Sec. 4, we compare the prediction accuracy of the model with some other classical models and analyze the prediction results by back-testing. Conclusions are provided in Sec. 5.

2. Data Processing Based on Feature Engineering

First of all, we select the Shanghai Composite Index (SCI), SZSE Component Index (SZSECI), SSE SME Composite (SSESC), CSI 300, CSI 500, and SSE 50,

which are the six major Chinese stock market indexes, with the day-level and minute-level price data from January 2009 to December 2020. The data is from Tushare, which is a big data open community in China. The reason why we choose these six indexes is that they have the characteristics of high equity stability, strong representativeness, and difficulty of human intervention. So that, they have a relatively stable trend and relatively high predictability, and they can be used as a measure of the overall emotional situation of the Chinese stock market with a relatively high correlation and similar trend (Jiang, 2014), which enables us to analyze these indexes together.

After obtaining a new stock index intraday price series every day, we need to classify the trend patterns of the day according to the series, so as to convert the numerical features based on price into the categorical features. For example, if the intraday trend pattern of the day is "high opening and low closing", the category is "1"; if the trend pattern is the shape of "V", then the category is "2"; if the trend pattern is the shape of inverted "V", the category is "3", etc. The specific types of trend patterns will be obtained adaptively by clustering historical data. By clustering the daily minute-level price series of each index in history, we can get the classification standard, which is called the standard class. Then, for the trend category of a new day, we can label the trend pattern category of the day by "nearest to a certain standard class", and get the features of the intraday trend pattern category.

2.1. Clustering in intraday minute-level price series

The clustering algorithm refers to the method of automatically dividing a pile of unlabeled data into several classes, which belongs to the unsupervised learning method. This method should ensure that the same class of data has similar characteristics. According to the distance or the similarity (closeness) between the samples, the more similar and less different samples are clustered into a class

(cluster), finally, several clusters are formed, so that the samples within the same cluster have high similarity and the differences between different clusters are high. The common clustering algorithms are K-means clustering, aggregation hierarchical clustering, mean-shift clustering and so on. In this paper, we choose k-means clustering, because it has the advantages of easy implementation, fast convergence and it has a better effect when the difference between clusters is obvious (Jain, 2010). K-means clustering algorithm organizes data objects into K partitions, $C = \{c_k, k = 1, 2, 3, \ldots, K\}$. Each partition represents a class c_k, and each class c_k has a class center μ_k. Although larger K can make the combination of categories more distinct, it also needs a large amount of data support to obtain higher classification accuracy, which is the limitation of our stock index dataset. Hence, we artificially select $K = 5$. Meanwhile, we use the correlation coefficient

as a criterion to judge the similarity and distance of the series data we studied, so as to classify the data with similar trend types into the same category (Li and Yao, 2007).

The steps of k-means algorithm in this research are as follows:

(1) First, a minute-level price dataset containing N data points of 240 dimensions is given (the stock market opens 240 minutes a day, with one close price data per minute). One of the data point is expressed as $P_i = \{p_{i,0}, p_{i,1}, p_{i,2}, \ldots, p_{i,239}\}$, where $0 \leq i < N$. Then the price data point is normalized so that only the price trend pattern information is retained to obtain the dataset for clustering, that is $X_i = \{x_{i,1}, x_{i,2}, x_{i,3}, \ldots, x_{i,240}\}$, where $x_{i,j} = \frac{p_{i,j} - \min\{P_i\}}{\max\{P_i\} - \min\{P_i\}}$. Then select K data points as initial class centers.
(2) Calculating the distance between each point in the dataset and the cluster center μ_k, and allocate them to the class represented by the most similar cluster center according to the nearest distance criterion.
(3) The mean value of all the data points in each category is calculated as the new cluster center of the class. The distance between all samples and new cluster centers is recalculated, and samples are reallocated according to the nearest distance criterion.
(4) Repeat steps (2) and (3) until cluster centers no longer change or reach the limit of iteration steps.

We selected the data between 2009.01.01 and 2017.12.31 of the six indexes as in-sample data for clustering. The clustering results of minute-level data in the day are shown in Fig. 1. The abscissa in the figure represents the time of every minute of trading in a day, and the ordinate is the normalized price value. It can be seen that there are obvious differences between each category. The five categories represent different trend types as well.

Only the information contained in the minute level trend characteristics of the day is far from enough to predict the trend of the next day. We select the trend

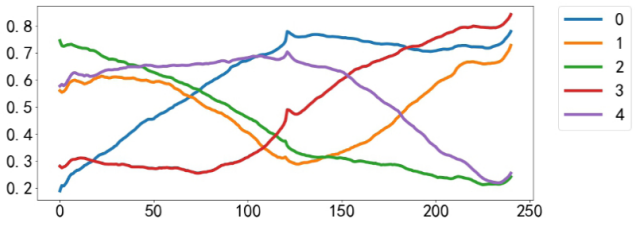

Fig. 1. Clustering result at minute-level of the last day.

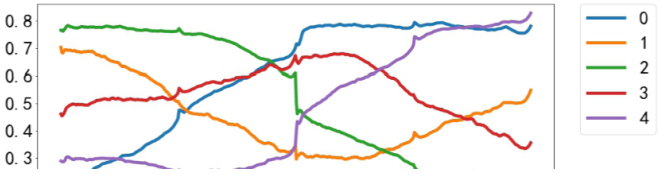

Fig. 2. Clustering result at minute-level of the last two days.

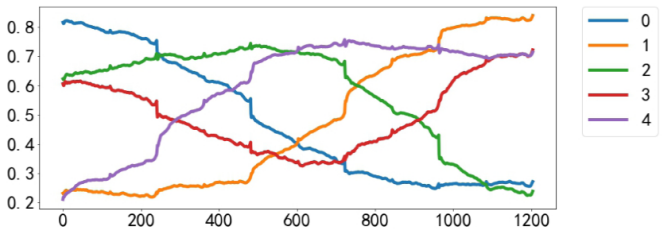

Fig. 3. Clustering result at minute-level of the last five days.

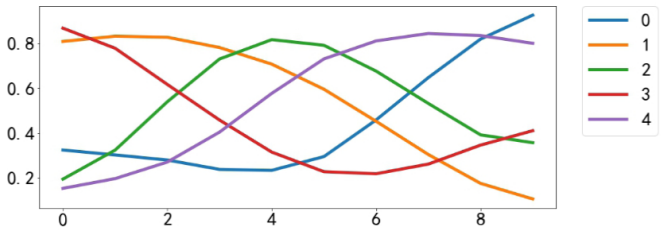

Fig. 4. Clustering result at day-level of the last 10 days.

characteristics under multiple time scale to form a feature combination. They are the minute-level trend pattern of the 2 days and 5 days, and the day-level trend pattern of the last 10 days, 20 days and 60 days. The clustering results of other time scales are shown in Figs. 2–6.

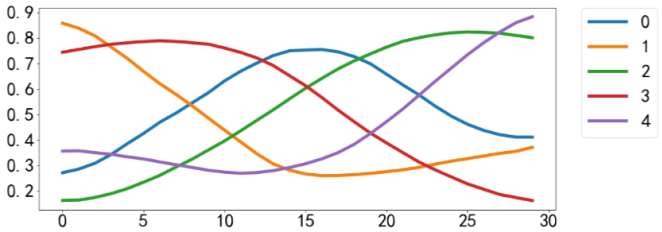

Fig. 5. Clustering result at day-level of the last 30 days.

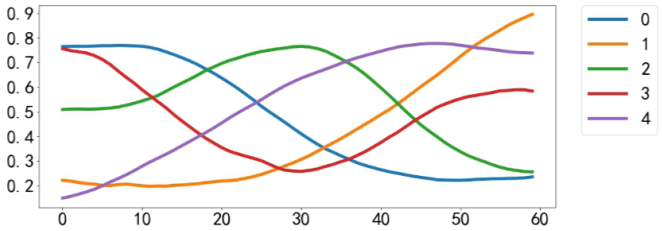

Fig. 6. Clustering result at day-level of the last 60 days.

2.2. The construction of features and targets

In addition to the above-mentioned six different time-scale trend features, adding the feature of the day of the week, we get seven types of features. For the new day's trend features, we will divide the trend series of the day into the cluster which is most similar to the series according to the correlation coefficient. Categories are represented by the numbers from "0" to "4" (it should be noted that although they are numbers, they have no numerical logic relationship. Different values only represent different categories). The prediction target Y is the price direction of the next day compared with that of the day. If the price rises, it is "1", otherwise it is "0". Finally, the format of the training dataset is summarized in Table 1.

3. Methodology

According to the categorical features of this research, we choose the CatBoost algorithm which can deal with the categorical features better. CatBoost was developed by Yandex Company in April 2017. Compared with other models based

Table 1. The format of the training dataset.

Variables	Value range	Type of value
Index name	—	string
Date	—	date time
Minute_1	["0","1","2","3","4"]	category
Minute_2	["0","1","2","3","4"]	category
Minute_5	["0","1","2","3","4"]	category
Day_10	["0","1","2","3","4"]	category
Day_30	["0","1","2","3","4"]	category
Day_60	["0","1","2","3","4"]	category
Weekday	["0","1","2","3","4","5","6"]	category
Y	["0","1"]	category

on the GBDT algorithm framework, such as XGBoost (Chen and Guestrin, 2016), CatBoost has the ability to better handle categorical features. At the same time, through the combination of categorical features, the feature dimensions are greatly enriched, and the information contained in categorical features can be fully mined.

Given the training set $S = \{(X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n)\}$, in which $X_i = (x_{i1}, x_{i2}, x_{i3}, \ldots, x_{im})$ is the m-dimensional vector containing numerical and categorical features, and Y_i is the prediction target.

First of all, CatBoost algorithm binarizes all numerical features: binary floating-point features, statistical information, and one-hot coding are binarized by using the oblivious tree as a base predictor.

Second, the categorical features are transformed into numerical features:

(1) Randomly arrange the samples to generate multiple random sequences.
(2) Given a certain sequence, the average tag value of the training dataset is used to replace the category:

$$x_{ik} = \frac{\sum_{j=1}^{n} I(x_{jk} = x_{ik}) \cdot Y_j}{\sum_{j=1}^{n} I(x_{jk} = x_{ik})}. \tag{1}$$

(3) Let $\theta = (\theta_1, \theta_2, \theta_n, \ldots, \theta_n)$, then convert the classified features into numerical values:

$$x_{\theta_p,k} = \frac{\sum_{j=1}^{p-1} I(x_{\theta_j,k} = x_{\theta_p,k}) \cdot Y_j + a \cdot P}{\sum_{j=1}^{p-1} I(_{\theta_j,k} = x_{\theta_p,k}) + P}. \tag{2}$$

Here, adding *a priori* value P and *a priori* weight a ($a > 0$), which helps to reduce the low-frequency noise.

Finally, when dealing with feature combination, CatBoost algorithm uses "greedy strategy" to combine:

(1) The first split of the tree does not carry out any combination.
(2) In the second split, all combinations and classification features existing in the current tree and all the classification features in the dataset are combined, and the combined value is immediately converted into a number.
(3) All splits selected in the tree are treated as classifications with two values and used in combination to produce a combination of numbers and classification features.

In the process of overcoming gradient deviation, CatBoost algorithm constructs a tree in two stages: (1) selecting the tree structure and calculating the value of leaf node after the tree structure is fixed; (2) enumerating different splitting methods, scoring the obtained tree by calculating the value of leaf node, so as to select the best segmentation. The values of the two-stage leaf nodes are calculated by using

best segmentation. The values of the two-stage leaf nodes are calculated by using the approximate values of gradient or Newton step size.

CatBoost achieves the synchronization of training dataset and processing categorical features, which greatly improves the efficiency of feature processing. Besides, the algorithm of calculating leaf nodes can effectively avoid over-fitting and reduce the need for super parameter optimization, which makes the model more universal.

4. Results and Analysis

4.1. Classification results based on CatBoost model

This research uses CatBoost algorithm to predict the trend of the stock market index based on Python. The data of six indexes between 2009.01.01 and 2017.12.31 are in-sample data for training, while the data between 2018.01.01 and 2020.12.01 are out-of-sample for predicting. For cross validation, we select 70% of the data in the sample as test data and 30% as validation data. For the parameters of the model, we fix the training rate at 0.1 first, then select different depths (from 1 to 16) and iterations (from 100 to 5000) as the parameters to be optimized, and test the optimal training results under different depths and iterations under the grid search method.

As shown in Figs. 7 and 8, when the model is iterated 788 times, the accuracy rate and the loss value of the testing set have reached the maximum and the minimum value, respectively, which are red points in figures. At this time, the accuracy of the training set is 68.5%, and the accuracy of the testing set is 62.2%. The corresponding parameters are listed in Table 2, which we regard as the optimal parameters.

Table 2. The optimal parameters of CatBoost.

Parameter	Optimal value
Iterations	788
Depth	4
Learning_rate	0.1
Loss_function	Logloss
Sub_sample	0.8
Eta	0.5
Min_child_weight	5
Eval_metric	AUC

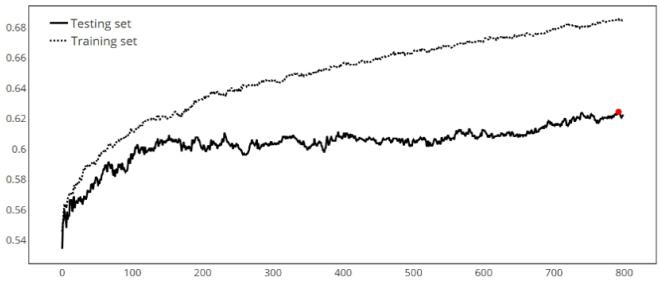

Fig. 7. Accuracy based on CatBoost algorithm.

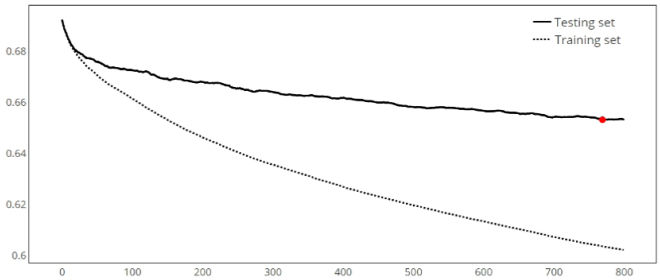

Fig. 8. Log loss value based on CatBoost algorithm.

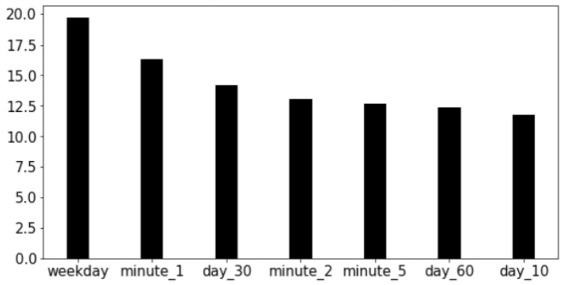

Fig. 9. Feature contribution ranking based on CatBoost algorithm.

Figure 9 shows the contribution of each feature to the model. All of these seven features contribute significantly to the model, and there is no invalid feature. Among them, the minute level trend pattern of the latest day and the day of the week contribute the most, which were 19.7% and 16.3%, respectively. Even the feature with the lowest contribution has the weight of 11.8%. This also verifies the day-of-the-week effect of the stock market and our expectation of the amount of information contained in the minute level trend.

Based on the model prediction results, we build a trading strategy and measure the accuracy of the prediction from the perspective of actual strategic profits. Here, we compare the long–short strategy and the long-only strategy separately, their trading rules are as follows:

(1) Long–short strategy: If it is predicted that the next day's closing price will be higher than today's, buy long today, if it will be lower, buy short today.
(2) Long-only strategy: If it is predicted that the next day's closing price will be higher than today's, buy long today, if it will be lower, close the position.

According to the out of sample prediction results, the back-testing results are shown in Table 3. It can be seen that from 2018.01 to 2020.12, short-only strategies all avoid many downside risks and win the index. Among them, SCI had the best maximum drawdown, 11.25%. When we apply the long–short strategy, the return of the strategy is almost doubled compared with that of the short-only strategy, which shows that the prediction effect of the model for the rise and fall is stable, and the excess return can be obtained. Among them, SSESC's long–short strategy gets the maximum annualized return of 52.50% and the best sharp ratio of 2.03.

Table 3. The back-testing results on CSI 300.

Strategy	Annualized return	Sharp ratio	Maximum drawdown
SCI Long–Short	34.71%	1.78	14.89%
SCI Long-Only	18.64%	1.21	**11.25%**
SCI Benchmark	2.57%	0.13	30.77%
SZSECI Long–Short	37.01%	1.49	16.57%
SZSECI Long-Only	23.65%	1.15	16.11%
SZSECI Benchmark	10.30%	0.41	38.92%
SSESC Long–Short	**52.50%**	**2.03**	24.29%
SSESC Long-Only	30.93%	1.47	15.57%
SSESC Benchmark	9.36%	0.36	41.04%
CSI 300 Long–Short	42.68%	2.00	16.78%
CSI 300 Long-Only	26.00%	1.47	14.34%
CSI 300 Benchmark	9.33%	0.43	32.46%
CSI 500 Long–Short	32.89%	1.34	30.98%
CSI 500 Long-Only	18.02%	0.94	23.61%
CSI 500 Benchmark	3.15%	0.13	37.66%
SSE 50 Long–Short	28.39%	1.34	17.06%
SSE 50 Long-Only	18.59%	1.13	14.63%
SSE 50 Benchmark	8.80%	0.41	28.87%

4.2. Comprehensive analysis

In this part, we compare XGBoost, lightGBM with CatBoost. In order to make the comparison more comprehensive, K-NearestNeighbor (KNN) and Multilayer

Perceptron (MLP), which are two classical classification machine learning algorithms, are also included. In order to deal with the feature of category, we use one-hot coding and update the distance function of categorical features in KNN. Finally, we average the predicting results on each index to get the average prediction performance of each model, which is shown in Table 4.

Comparing CatBoost with other models, it can be found that although the improvement of prediction accuracy is small, CatBoost has more stable returns. The reason is that on one hand, profits are growing exponentially, a little

Table 4. Comparison of average prediction performance of each model.

Model	Prediction accuracy	Average annualized return	Average maximum drawdown
XGBoost	0.53	21.55%	24.75%
LightGBM	0.52	16.69%	22.87%
CatBoost	0.55	38.03%	20.10%
KNN	0.52	18.13%	28.21%
MLP	0.51	6.62%	31.33%

improvement can lead to a big accumulated gap; on the other hand, the back-testing results are not only based on the trend but also based on the range of price fluctuations, Catboost can better grasp the trend and give more accurate predictions in more valuable time periods.

4.3. Rolling back-testing results

In order to compare the real prediction effect of CatBoost in different periods of the past, we use the rolling back-testing method to simulate the real environment and simulate the return of strategy in the historical real trading environment. The rolling back-testing process is as follows:

(1) Assuming that the strategy starts in 2012.01.01. At the beginning of each year, the historical data from 2009.01.01 to the day t is used to cluster, then we get the feature sample X_t and target set Y_t of all historical data. After that, we use CatBoost to train the sample set and get the model M_t.
(2) Since the beginning of the year, the feature data of the day i are calculated at the close of the stock market to predict the trend by model M_t. Then we adjust the position of the day i according to the predicting results (note that there is an assumption that we can still trade at the close of the day and there are only 11 months of data in 2020).

Table 5. The rolling back-testing results.

Strategy	Annualized return	Sharp ratio	Maximum drawdown
SCI Long–Short	32.10%	1.49	21.88%
SCI Long-Only	19.83%	1.21	27.52%
SCI Benchmark	7.57%	0.35	52.30%
SZSECI Long–Short	33.35%	1.29	37.51%
SZSECI Long-Only	21.00%	1.07	30.59%
SZSECI Benchmark	8.65%	0.33	60.83%
SSESC Long–Short	**40.85%**	**1.53**	26.60%
SSESC Long-Only	26.67%	1.34	26.58%
SSESC Benchmark	12.49%	0.47	61.75%
CSI 300 Long–Short	27.34%	1.16	26.10%
CSI 300 Long-Only	19.60%	1.14	21.61%
CSI 300 Benchmark	11.86%	0.50	44.70%
CSI 500 Long–Short	33.58%	1.46	30.21%
CSI 500 Long-Only	22.60%	1.28	23.84%
CSI 500 Benchmark	11.62%	0.50	46.70%
SSE 50 Long–Short	36.87%	1.38	30.32%
SSE 50 Long-Only	24.22%	1.21	28.32%
SSE 50 Benchmark	11.56%	0.43	65.20%

Finally, annual statistics of the strategy under rolling back-testing are shown in Table 5. Although the annualized return is lower than that in the fixed period, it

still remains at a stable high level. The SSESC's long–short strategy still has the best performance with an annualized return of 40.85% and a sharp ratio of 1.53.

5. Conclusion

In this paper, we take the day trend direction of the Chinese stock market index as the research object, innovatively uses the interday minute-level price series to construct the features of the interday price trend categories. Combined with different time scales trend categories and the day-of-the-week effect, a feature combination containing price trend information is constructed.

In order to predict the datasets of categorical features better, we use the CatBoost model to build an index trend prediction model. Experimental results show that the out-of-sample prediction accuracy is 0.55, and the long–short trading strategy based on prediction results gets an average annualized return of 38.03% with an average maximum drawdown of 20.10%. The performance advantages of this model are compared with other classical classification models comprehensively. We find that the prediction accuracy of CatBoost is only a little higher than that of other models, but the annualized return of back-testing is greatly improved. Moreover, the rolling back-testing, from 2012 to 2020, shows that the strategy based on the CatBoost model has a good and stable performance in each historical period of each index, with effective forecasting performance and high profitability. Therefore, the trading strategy based on the feature engineering and model constructed in this paper has great investment potential.

References

Al-Sarem, M, F Saeed, W Boulila, AH Emara, M Al-Mohaimeed and M Errais (2020). Feature selection and classification using CatBoost method for improving the performance of predicting Parkinson's disease, *Advances on Smart and Soft Computing*, 2020, 189–199.

Bai, Y (2009). Forecast and analysis of shanghai stock index based upon arima model, *Science Technology and Engineering*, 009(016), 4885–4888.

Ballings, M, D Van den Poel, N Hespeels and R Gryp (2015). Evaluating multiple classifiers for stock price direction prediction, *Expert Systems with Applications*, 42, 7046–7056.

Brock, W, J Lakonishok and B Lebaron (1992). Simple technical trading rules and the stochastic properties of stock returns, *The Journal of Finance*, 47(5), 1731–1764.

Caginalp, G and M Desantis (2011). Nonlinearity in the dynamics of financial markets, *Nonlinear Analysis Real World Applications*, 12, 1140–1151.

Cai, J, LI Yuming and QI Yuehua (2006). The day-of-the-week effect: New evidence from the Chinese stock market, *Chinese Economy*, 39(2), 71–88.

Chen, T and C Guestrin (2016). XGBoost: A scalable tree boosting system, *The 22nd ACM SIGKDD International Conference*, ACM, pp. 785–794.

Dorogush, AV, V Ershov and A Gulin (2018). Catboost: Gradient boosting with categorical features support, arXiv:1810.11363.

Dou, X (2020). Online purchase behavior prediction and analysis using ensemble learning. *2020 IEEE 5th International Conference on Cloud Computing and Big Data Analytics (ICCCBDA)*, IEEE, pp. 532–536.

Guan, L and M Zhao (2005). Markov chain model prediction of shanghai composite index trend, *Journal of Shandong Academy of Governance*, 69, 95–96.

Hao, Y and Q Gao (2020). Predicting the trend of stock market index using the hybrid neural network based on multiple time scale feature learning, *Applied Sciences*, 10(11), 3961.

Hassan, MR, B Nath and M Kirley (2007). A fusion model of hmm, ann and ga for stock market forecasting, *Expert Systems with Applications*, 33, 171–180.

Huang, W, Y Nakamori and S-Y Wang (2005). Forecasting stock market movement direction with support vector machine, *Computers & Operations Research*, 32, 2513–2522.

Jain, AK (2010). Data clustering: 50 years beyond k-means, *Pattern Recognition Letters*, 31(8), 651–666.

Jiang G. (2014). Correlation analysis and prediction of Shanghai and Shenzhen stock market price index, Doctoral dissertation, Jinan University.

Kang, P, Z Lin, S Teng, G Zhang and W Zhang (2019). Catboost-based Framework with Additional User Information for Social Media Popularity Prediction. *The 27th ACM International Conference*, ACM, pp. 2677–2681.

Li, L and D Yao (2007). A new method of spatio-temporal topographic mapping by correlation coefficient of k-means cluster, *Brain Topography*, 19(4), 161–176.

Malkiel, BG and EF Fama (1970). American finance association efficient capital markets: A review of theory and empirical work, *Journal of Finance*, 25(2), 383–417.

Marshall, BR and RH Cahan (2005). Is technical analysis profitable on a stock market

which has characteristics that suggest it may be inefficient?, *Research in International Business & Finance*, 19(3), 384–398.
Ni, J, L Zhang, J Tao and X Yang (2020). Prediction of stocks with high transfer based on ensemble learning, *Journal of Physics: Conference Series*, 1651(1), 012124.
Nti, IK, AF Adekoya and BA Weyori (2020). A comprehensive evaluation of ensemble learning for stock-market prediction, *Journal of Big Data*, 7(1), 1–40.
Patel, J, S Shah, P Thakkar and K Kotecha (2015). Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques. *Expert Systems with Applications*, 42(1), 259–268.

Postnikov, EB, DA Esmedljaeva and AI Lavrova (2020). A CatBoost machine learning for prognosis of pathogen's drug resistance in pulmonary tuberculosis. *2020 IEEE 2nd Global Conference on Life Sciences and Technologies (LifeTech)*. IEEE, pp. 86–87.
Prokhorenkova, L, G Gusev, A Vorobev, AV Dorogush and A Gulin (2018). CatBoost: Unbiased boosting with categorical features. In *Advances in Neural Information Processing Systems*, pp. 6638–6648.
Punmiya, R and S Choe (2019). Energy theft detection using gradient boosting theft detector with feature engineering-based preprocessing, *IEEE Transactions on Smart Grid*, 10(2), 2326–2329.