Machine Learning in Stock Price Forecast

Zhen Sun[1] Shangmei Zhao[1]

[1]School of Economics and Management Beihang University Beijing, China

Abstract—This paper analyzed and compared the forecast effect of three machine learning algorithms (multiple linear regression, random forest and LSTM network) in stock price forecast using the closing price data of NASDAQ ETF and data of statistical factors. The test results show that the prediction effect of the closing price data is better than that of statistical factors, but the difference is not significant. Multiple linear regression is most suitable for stock price forecast. The second is random forest, which is prone to overfitting. The forecast effect of LSTM network is the worst and the values of RMSE and MAPE were the highest. The forecast effect of future stock price using closing price of NASDAQ ETF is better than that using statistical factors, but the difference is not significant.

1 Introduction

Machine learning [1] is the general term for a class of algorithms that attempt to extract hidden information from historical data and solve the problems of prediction or classification. At present, the research of machine learning in stock prediction mainly focus on two key aspects. One is that researchers improved the original algorithm and try to prove that the prediction effect of the improved algorithm is better than that of original algorithm for the future stock price [2,3]. The other is researchers analyze and verify which algorithm is the most suitable for the forecast of future stock price based on compassion of several machine learning algorithms [4,5,6,7,8,9]. The most commonly used algorithms in the stock prediction are neural network algorithms that include multiple linear regression [10] and Long Short Term Memory [11], support vector machines [12] and random forest [13;14]. The remaining algorithms [15] are usually used as complementary algorithms to improve and optimize the above algorithms.

2 Machine Learning Algorithms

2.1 Multiple Linear Regression

Multiple linear regression is also an algorithm that aims to calculate the parameters (w, b). The specific steps are as follows. Firstly, we need to design a multiple linear regression mode, $\hat{Y} = w^T X + b$; Secondly, we will construct the objective function, $Obj(w,b) = \arg min_{w,b} \mathcal{L}(Y, \hat{Y}) = \arg min_{w,b} \sum (Y_i - \hat{Y}_i)^2$, using the least squares method; Finally, the gradient descent method is used to train the objective function and the parameters of model (w, b) are obtained. Once the model is trained, it can be used to make relevant predictions.

2.2 Random Forest

Random forest algorithm can solve the classification and regression problems. In general, the Random Forest uses the Bootstrapping algorithm to construct N training sets from the sample data. It trains the sample data by the individual classification algorithm to obtain N individual classifiers. If we want to solve the classification problems, the final classification result is voted on by the N individual classifiers. If we want to solve the regression problems, the final classification result is the mean value of outputs of the N individual classifiers. The random uniform sampling is used in the Random Forest algorithm. The weights of individual classifier are equals and each individual classifier can be generated in parallel. We choose Classification And Regression Tree (CART) as the algorithm of individual classifier in this paper.

2.3 LSTM

LSTM (Long Short Term Memory) is a type of recurrent neural network (RNN) that can solve the problem of long-term dependencies and can apply the previous information to the current task.

(1) The related concepts of LSTM

1) The cell state value C_i. C_i is the *i-th* cell state value. The process from state C_i to state C_{i+1} is called the $(i+1)-th$ process.

2) The output of function σ is a vector. Each element of σ is a number between 0 and 1. σ represent the weight

that can decide the retained number of input information. 0 means "Don't let any information pass" and 1 means "Let all information pass".

3) The output of function $tanh$ is a vector. Each element of $tanh$ is a number between -1 and 1. However, $tanh$ is different from σ. It maps the value of the input information to the value between -1 and 1. It just change the form of input information and remain the property of input information.

(2) The mathematical expression of LSTM

Firstly, the four variables that are f_t, i_t, out_t and $\tilde{\mathrm{C}}_\mathrm{t}$ are calculated in LSTM network.

1) $f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right)$, where h_{t-1} represents the output of the $(t-1)-th$ process, x_t represents the input of the $t-th$ process. W_f and b_f represent the weight and bias.

2) $i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$, where h_{t-1} represents the output of the $(t-1)-th$ process, x_t represents the input of the $t-th$ process. W_i and b_i represent the weight and bias.

3) $out_t = \sigma(W_{out} \cdot [h_{t-1}, x_t] + b_{out})$, where h_{t-1} represents the output of the $(t-1)-th$ process, x_t represents the input of the $t-th$ process. W_{out} and b_{out} represent the weight and bias.

4) $\tilde{C}_t = tanh(W_{\tilde{C}} \cdot [h_{t-1}, x_t] + b_{\tilde{C}})$, where h_{t-1} represents the output of the $(t-1)-th$ process, x_t represents the input of the $t-th$ process. $W_{\tilde{C}}$ and $b_{\tilde{C}}$ represent the weight and bias.

Secondly, the cell state value C_i are calculated.

$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$, where f_t represent the weight of C_{t-1} and i_t represent the weight of $\tilde{C}_t$.

Finally, the mathematical expression of LSTM network can be obtained.

The mathematical expression of LSTM network is: $h_t = out_t * tanh(C_t)$. h_t represents the output of the $t-th$ process, out_t is the weight and can decide the remained number of C_t.

3 Data description

3.1 Standardized method of input data

In this paper, we used "Z-score" statistical method to transform the original data $\mathrm{X} = [\mathrm{x_1}, \mathrm{x_2}, \cdots, \mathrm{x_n}]$. This formula can be expressed as: $\mathrm{Y} = \frac{\mathrm{X}-\bar{\mathrm{X}}}{\mathrm{S}}$. Where, $\bar{X} = \frac{1}{n}\sum_{i=1}^{n} x_i$, $S = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2}$, the mean value of Y is 0 and the variance value of Y is 1.

3.2 The types of input data

There are two types of input data in this paper, which are as follows.

Firstly, the first type of is the standardized data of the closing price of NASDAQ ETF. This type of data can be described as: $D_{first} = \left(X_{first}, Y_{first}\right)$. $X_{first} = [X_1, \cdots, X_i, \cdots]$, where $X_i = [x_{i-n}, \cdots, x_{i-1}]$, X_i represents the input data of the $i-th$ day, x_{i-n} represents the closing price of the $(i-n)-th$ day, $n = 10$. $Y_{first} = [Y_1, \cdots Y_i, \cdots]$, where Y_i represents the output value of the $i-th$ day. The total number of data is 776. The time period of data is from June 1, 2016 to June 28, 2019. The number of training data accounted for 80 percent of total number. The number is 620. The time period of training data is from June 1, 2016 to November 12, 2018. The test data is divided into in-sample data and out-of-sample data. The in-sample data is selected randomly from training data, it accounted for 20 percent of total number. The out-of-sample data is the remained data that means the training data are removed form total data, it accounted for 20 percent of total number. The time period of out-of-sample data is from November 13, 2018 to June 28, 2019 and the number of out-of-sample data is 156.

Secondly, the second type of data is the standardized data of statistical factors that can be constructed by closing price, opening price, maximum price, minimum price and trading volume data of NASDAQ ETF. This type of data can be described as: $\mathrm{D_{second}} = (\mathrm{X_{second}}, \mathrm{Y_{second}})$. $\mathrm{X_{second}} = [\mathrm{X_1}, \cdots, \mathrm{X_i}, \cdots]$, where $\mathrm{X_i} = \left[\mathrm{x_{i1}}, \mathrm{x_{i2}}, \cdots, \mathrm{x_{ij}}, \cdots \mathrm{x_{in}}\right]$, X_i represents the input data of the $i-th$ day. x_{in} represents statistical factors of the $j-th$ statistic factor in the $i-th$ day, $n = 55$. $Y_{second} = [Y_1, \cdots Y_i, \cdots]$, where Y_i represents the output value of the $i-th$ day. The total number of data of statistical factor is 20075. The time period of data is from January 17, 2018 to June 28, 2019. The number of training data accounted for 80% of total number. The number is 16,115. The time period of training data is from January 17, 2018 to March 18, 2019. The test data is divided into in-sample data and out-of-sample data. The in-sample data is selected randomly from training data, it accounted for 20 percent of total number. The out-of-sample data is the remained data that means the training data are removed form total data, it accounted for 20 percent of total number. The time period of out-of-sample data is from March 19, 2019 to June 28, 2019 and the number of out-of-sample data is 3960.

4 Comparison of forecasting algorithms

4.1 Training model

Through training three classifiers (Multiple linear regression, LSTM networks, Random Forest) respectively by the training set D_{train}, we can obtain the stable parameters of three models. The detailed steps are as follows.

Firstly, we process the original training data X_{train} by the classifier, and obtain the prediction value $\hat{Y}_{train}$;

Secondly, we compare the prediction value $\hat{Y}_{train}$ with the label value Y_{train} and update the parameters through repetitive iteration.

4.2 Assessment model

We use the two evaluation indicators that are RMSE and MAPE to analyze the forecast effect of the three algorithms and determine which algorithm has the better forecast effect. The construction methods of the two indicators are as follows:

$$(1)\ RMSE:\ RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(t_i - p_i)^2}$$

$$(2)\ MAPE:\ MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{t_i - p_i}{t_i}\right|$$

Where, p_i represents the predicted value of the $i-th$ day, and t_i represents the true value of the $i-th$ day. For these two indicators, the lower the value, the better the prediction effect of the model.

5 Forecasting result based on closing price of NASDAQ ETF

5.1 Forecasting result of the test within the samples

TABLE I. FORECASTING RESULT OF THE TEST WITHIN THE SAMPLES

Forecasting result (The test within the samples)		
	RMSE	MAPE
Multiple Linear Regression	0.051	0.387
Random Forest	0.036	0.144
LSTM networks	1.420	2.585

When we forecast the stock price using closing price data of NASDAQ ETF within the samples, the result shows that the RMSE of multiple linear regression is 0.051 and the MAPE of multiple linear regression is 0.387. The RMSE of random forest is 0.036 and the MAPE of random forest is 0.144. The RMSE of LSTM networks is 1.420 and the MAPE of LSTM networks is 2.585. Therefore, we can see that random forest get the best forecast effect, the values of RMSE and MAPE are the lowest. LSTM networks achieve the worst forecast effect, the values of RMSE and MAPE are the highest. The forecast effect of multiple linear regression is between random forest and LSTM networks, the values of RMSE and MAPE are close to random forest.

5.2 Forecasting result of the test outside the samples

TABLE II. FORESTING RESULT OF THE TEST OUTSIDE THE SAMPLES

Foresting result (The test outside the samples)		
	RMSE	MAPE
Multiple Linear Regression	0.236	0.423
Random Forest	0.283	0.561
LSTM networks	1.399	3.215

When we forecast the stock price using closing price data of NASDAQ ETF outside the samples, the result shows that the RMSE of multiple linear regression is 0.236 and the MAPE of multiple linear regression is 0.423. The RMSE of random forest is 0.283 and the MAPE of random forest is 0.561. The RMSE of LSTM networks is 1.399 and the MAPE of LSTM networks is 3.215. Therefore, we can see that multiple linear regression get the best forecast effect, the values of RMSE and MAPE are the lowest. LSTM networks achieve the worst forecast effect, the values of RMSE and MAPE are the highest. The forecast effect of random forest is between multiple linear regression and LSTM networks, the values of RMSE and MAPE are close to multiple linear regression.

6 Forecasting result based on the data of statistical factors

6.1 Forecasting result of the test within the samples

TABLE III. FORECASTING RESULT OF THE TEST WITHIN THE SAMPLES

Forecasting result (The test within the samples)		
	RMSE	MAPE
Multiple Linear Regression	0.095	0.735
Random Forest	0.043	0.212
LSTM networks	1.199	5.256

When we forecast the stock price using the data statistical factors within the samples, the results show that the RMSE of multiple linear regression is 0.095 and the MAPE of multiple linear regression is 0.735. The RMSE of random forest is 0.043 and the MAPE of random forest is 0.212. The RMSE of LSTM networks is 1.199 and the MAPE of LSTM networks is 5.256. Therefore, we can see that random forest get the best forecast effect, the values of RMSE and MAPE are the lowest. LSTM networks achieve the worst forecast effect, the values of RMSE and MAPE are the highest. The forecast effect of multiple linear regression is between random forest and LSTM networks, the values of RMSE and MAPE are close to random forest.

6.2 Forecasting result of the test outside the samples

TABLE IV. FORESTING RESULT OF THE TEST OUTSIDE THE SAMPLES

Foresting result (The test outside the samples)		
	RMSE	MAPE
Multiple Linear Regression	0.259	0.169
Random Forest	0.362	0.309
LSTM networks	1.126	5.241

When we forecast the stock price using the data of statistical factors outside the samples, the results show that the RMSE of multiple linear regression is 0.259 and the MAPE of multiple linear regression is 0.169. The RMSE of random forest is 0.362 and the MAPE of random forest is 0.309. The RMSE of LSTM networks is 1.126 and the MAPE of LSTM networks is 5.241. Therefore, we can see that multiple linear regression get the best forecast effect, the values of RMSE and MAPE are the lowest. LSTM networks achieve the worst forecast effect, the values of RMSE and MAPE are the highest. The forecast effect of random forest is between multiple linear regression and LSTM networks, the values of RMSE and MAPE are close to multiple linear regression.

7 Conclusion

Three supervised learning algorithms (Multiple linear regression, Random forest and LSTM networks) are used in this study to forecast the future stock price using closing price of NASDAQ ETF and statistical factors. The forecast results of the three algorithms show that multiple linear regression is more suitable for the prediction of future stock price. The random forest has the best prediction effect for the in-sample data, but the prediction effect for the out-of-sample data is not as good as the multiple linear regression, which means the random forest is prone to overfitting. LSTM networks both in-sample and out-of-sample data get the worst forecast effect, the values of RMSE and MAPE were the highest. The forecast effect of future stock price using closing price of NASDAQ ETF is better than that using statistical factors, but the difference is not significant.

8 Statistical factors

Factor1: $\sigma(vol)_i = \sqrt{\sum_{j=i-N+1}^{i}\left(vol_j - mean\left(\left[vol_{i-N+1},\cdots,vol_j,\cdots vol_i\right]\right)/(N-1)\right)}$, *where* vol_j *is the trading volume of the* $j-th$ *day,* $N = 10$.

Factor2: $SMA(vol_t, N, M) = Y_t = (M * vol_t + (N-M) * Y_{t-1})/N$, *where* $t \geq 2$, $N = 10$, $M = 1$.

Factor3: $\sigma(close)_i = \sqrt{\sum_{j=i-N+1}^{i}\left(close_j - mean\left(\left[close_{i-N+1},\cdots,close_j,\cdots close_i\right]\right)/(N-1)\right)}$, *where* $close_j$ *is the closing price of the* $j-th$ *day,* $N = 10$.

Factor4: $RSV_i = \frac{max\left([close_{i-N+1},\cdots,close_i]\right)-close_i}{max\left([close_{i-N+1},\cdots,close_i]\right)-min\left([close_{i-N+1},\cdots,close_i]\right)} \times 100$, $WR_i = SMA(RSV_i, N, M)$, *where* $N \geq i \geq 1$, $N = 10$, $M = 1$.

Factor5: If $close_i > close_{i-1}$, $DIF_i = close_i - min\left([low_i, close_{i-1}]\right)$. *If* $close_i < close_{i-1}$, $DIF_i = close_i - max\left([high_i, close_{i-1}]\right)$, $ACD_i = \sum_{j=i-N+1}^{i} DIF_j$, *where* $close_i$ *is the closing price of the* $i-th$ *day,* low_i *is the lowest price of the* $i-th$ *day,* $high_i$ *is the highest price of the* $i-th$ *day,* $N = 10$.

Factor6: If $open_i < open_{i-1}$, $DBM_i = max[(open_i - low_i), (open_{i-1} - open_i)]$, *else* $DBM_i = 0$. *If* $open_i > open_{i-1}$, $DTM_i = max[(high_i - low_i), (open_i - open_{i-1})]$, *else* $DTM_i = 0$. $STM_i = \sum_{j=i-N-1}^{i} DTM_j$. $SBM_i = \sum_{j=i-N-1}^{i} DBM_j$. *If* $STM_i > SBM_i$, $ADTM_i = (STM_i - SBM_i)/STM_i$. *If* $STM_i < SBM_i$, $ADTM_i = (STM_i - SBM_i)/SBM_i$. *If* $STM_i = SBM_i$, $ADTM_i = 0$. *Where* $open_i$ *is the opening price of the* $i-th$ *day,* $N = 10$.

Factor7: $CLV_i = \frac{(close_i - low_i) - (high_i - close_i)}{(high_i - low_i)}$, $ADV_i = ADV_{i-1} + vol_i * CLV_i$.

Factor8: $AR_i = \frac{\sum_{j=i-N+1}^{i}(high_j - open_j)}{\sum_{j=i-N+1}^{i}(open_j - low_j)}$, $BR_i = \frac{\sum_{j=i-N+1}^{i}(max[0,(high_j - close_{j-1})])}{\sum_{j=i-N+1}^{i}(max[0,(close_{j-1} - low_j)])}$, *where* $N = 10$

Factor9: $ARC = SMA\left(\frac{close_i}{close_{i-N+1}}, N, 1\right)$, *where* $N = 10$.

Factor10: $Aroon\_up_i = [(N - HF_i)/N] * 100$, $Aroon\_down = [(N - LF_i)/N] * 100$, $Aroon = Aroon\_up - Aroon\_down$. *Where* HF_i *represent the number from* $close_h$ *to* $close_i$, LF_i *represent the number from* $close_l$ *to* $close_i$, $close_l = min([close_{i-N+1}, \cdots, close_i])$, $close_h = max([close_{i-N+1}, \cdots, close_i])$, $N = 10$.

Factor11: $AA_i = abs(high_i - close_{i-1})$, $BB_i = abs(low_i - close_{i-1})$, $CC_i = abs(high_i - low_{i-1})$, $DD_i = abs(close_{i-1} - open_{i-1})$. *If* $AA_i > max(BB_i, CC_i)$, $R_i = AA_i + \frac{BB_i}{2} + \frac{DD_i}{4}$; *If* $AA_i \leq max(BB_i, CC_i)$ *and* $BB_i > max(CC_i, AA_i)$, $R_i = BB_i + \frac{AA_i}{2} + \frac{DD_i}{4}$; *If* $AA_i \leq max(BB_i, CC_i)$ *and* $BB_i \leq max(CC_i, AA_i)$, $R_i = CC_i + \frac{DD_i}{4}$, $X_i = (close_i - close_{i-1}) + \frac{1}{2}(close_i - open_i) + (close_{i-1} - open_{i-1})$, $SI_i = 16 * \frac{X_i}{R_i} * max(AA_i, BB_i)$, $ASI_i = \sum_{j=i-N+1}^{i} SI_j$, *where* $N = 10$, $abs(X) = |X|$.

Factor12: $TR_i = max[(high_i - low_i), |close_{i-1} - high_i|, |close_{i-1} - low_i|]$, $ATR_i = MA(TR_i, N)$, *where* $N = 10$, $MA(TR_i, N) = mean([TR_{i-N+1}, \cdots, TR_i])$.

Factor13: $BBI_i = mean\left(\left[MA(close_j, M1), MA(close_j, M2), MA(close_j, M3), MA(close_j, M4)\right]\right)$, *where* $M1 = 3$, $M2 = 6$, $M3 = 12$, $M4 = 25$.

Factor14: $BIAS_i = \frac{\left(close_i - MA(close_j, N)\right)}{MA(close_j, N)} * 100$, *where* $N = 10$.

Factor15: $TYP_i = mean(high_i + low_i + close_i)$, $AVEDEV_i = \sum_{j=i-N+1}^{i}\left|TYP_i - MA(close_j, N)\right|$, $CCI_i = \left(TYP_i - MA(close_j, N)\right) / 0.015 * AVEDEV_i$, *where* $N = 10$.

Factor16: $BV_i = vol_i * \frac{[(close_i - low_i) - (high_i - close_i)]}{(high_i - low_i)}$, $Chaikin\ Oscillator = EMA(BV_i, 10) - EMA(BV_i, 3)$, *where* $EMA(X_t, N) = Y_t = \frac{1}{N} * X_t + \left(1 - \frac{1}{N}\right) * Y_{t-1}$.

Factor17: $CV = 100 * \frac{\left(EMA((high_i - low_i), N) - EMA((high_{i-N+1} - low_{i-N+1}), N)\right)}{EMA((high_{i-N+1} - low_{i-N+1}), N)}$, *where* $N = 10$.

Factor18: If $close_i > close_{i-1}$, $CZ1 = close_i - close_{i-1}$. *If* $close_i < close_{i-1}$, $CZ2 = |close_i - close_{i-1}|$. $SU_j = \sum_{j-N+1}^{i} CZ1_j$, $SD_j = \sum_{j-N+1}^{i} CZ2_j$. $CMO_i = 100 * [(SU_i - SD_i)/(SU_i + SD_i)]$, *where* $N = 10$.

Factor19: $R1_i = \frac{(close_i - close_{i-n1+1})}{close_{i-n1+1}} * 100$, $R2_i = \frac{(close_i - close_{i-n2+1})}{close_{i-n2+1}} * 100$, $RC_i = R1_i + R2_i$, $Coppock_i = WMA(RC_i, n3)$, *where* $WMA(X_i, N) = \frac{1*X_{i-N+1} + 2*X_{i-N+2} + \cdots + NX_i}{1+2+\cdots+N}$, $n1 = 5$, $n2 = 10$, $n3 = 10$.

Factor20: $MID_i = mean(high_i + low_i + close_i)$, $CR_i = \frac{\sum_{j=i-N+1}^{i}(max[0,(high_j - MID_{j-1})])}{\sum_{j=i-N+1}^{i}(max[0,(MID_{j-1} - low_j)])} * 100$, *where* $N = 10$.

Factor21: $BIAS_i = \left(close_i - MA(close_j, N)\right) / MA(close_j, N)$, $DIF_i = (BIAS_i - BIAS_{i-M+1})$, $DBCD_i = SMA(DIF_i, T, 1)$, *where* $N = 10$, $M = 5$, $T = 10$。

Factor22: If $(high_i + low_i) \le (high_{i-1} + low_{i-1})$, $DMZ_i = 0$. *If* $(high_i + low_i) > (high_{i-1} + low_{i-1})$, $DMZ_i = max[|high_i - high_{i-1}|, |low_i - low_{i-1}|]$. *If* $(high_i + low_i) \ge (high_{i-1} + low_{i-1})$, $DMZ_i = 0$. *If* $(high_i + low_i) < (high_{i-1} + low_{i-1})$, $DMF_i = max[|high_i - high_{i-1}|, |low_i - low_{i-1}|]$. $DIZ_i = \sum_{j=i-N+1}^{i}(DMZ_j) / \left(\sum_{j=i-N+1}^{i}(DMZ_j) + \sum_{j=i-N+1}^{i}(DMF_j)\right)$, $DIF_i = \sum_{j=i-N+1}^{i}(DMF_j) / \left(\sum_{j=i-N+1}^{i}(DMF_j) + \sum_{j=i-N+1}^{i}(DMF_j)\right)$. $DDI_i = DIZ_i - DIF_i$. *Where* $N = 10$.

Factor23: $A_i = high_i - low_i$, $B_i = abs(high_i - close_{i-1})$, $C_i = abs(low_i - close_{i-1})$, $TR_i = \sum_{j=i-N+1}^{i}(max([A_i, B_i, C_i]))$. *If* $HD_i > LD_i$ *and* $HD_i > 0$, $HD_i = high_i - high_{i-1}$; *else* $HD_i = 0$. $DMP_i = \sum_{j=i-N+1}^{i}(HD_i)$, $PDI_i = (DMP_i / TR_i) * 100$. *If* $LD_i > HD_i$ *and* $LD_i > 0$, $LD_i = low_{i-1} - low_i$; *else* $LD_i = 0$. $DMM_i = \sum_{j=i-N+1}^{i}(LD_i)$, $MDI_i = (MDD_i / TR_i) * 100$. $ADX_i = \sum_{j=i-N+1}^{i}\left(\frac{abs(MDI_j - PDI_j)}{(MDI_j + PDI_j)} * 100\right)$, $ADXR_i = (ADX_i + ADX_{i-N+1})/2$. *Where* $N = 10$, $abs(c) = |c|$.

Factor24: $EMV_i = EMA\left(\left(\frac{(high_i + low_i)}{2} - \frac{(high_{i-1} + low_{i-1})}{2}\right) * \frac{(high_i - low_i)}{vol_i}, N\right)$, *where* $N = 10$.

Factor25: $rise_i = |high_i - EMA(close_i, N)|$, $down_i = |low_i - EMA(close_i, N)|$, $Elder_i = (rise_i - down_i)/close_i$, *where* $N = 10$.

Factor27: $RSV_i = \frac{(close_i - min(low_{i-n+1}, \cdots, low_i))}{(max(high_{i-N+1}, \cdots, high_i) - min(low_{i-N+1}, \cdots, low_i))} * 100$, $K_i = \frac{2}{3} * K_{i-1} + \frac{1}{3} * RSV_i$, $D_i = \frac{2}{3} * D_{i-1} + \frac{1}{3} * K_i$, $J_i = 3 * K_i - 2 * D_i$, *where* $N = 10$. $K_1 = D_1 = 50$.

Factor26: $M_i = (close_i + high_i + low_i)/3$, *if* $M_i > M_{i-1}$, $vol_i = vol_i$; *if* $M_i < M_{i-1}$, $vol_i = -vol_i$. $vol1_i = EMA(vol_i, n1)$, $vol2_i = EMA(vol_i, n2)$, $KO_t = EMA((vol1_i - vol2_i), n3)$, *where* $n1 = 5$, $n2 = 10$, $n2 = 5$.

Factor28: $MC_i = \frac{MA(close_i, N)}{close_i}$, *where* $N = 10$.

Factor29: $close1_i = EMA(close_i, n1)$, $close2_i = EMA(close_i, n2)$, $DIFF_i = close1_i - close2_i$, $DEA_i = EMA(DIFF_i, m)$, $MACD_i = 2 * (DIFF_i - DEA_i)$, *where* $n1 = 5$, $n1 = 10$, $m = 5$.

Factor30: $MassFlowIndex = \frac{EMA(high_i - low_i, N)}{EMA(EMA(high_i - low_i, N), N)}$, *where* $N = 10$.

Factor31: $TYP_i = (high_i + low_i + close_i)/3$. *If* $TYP_i > TYP_{i-1}$, $A_i = TYP_i * vol_i$; *else* $A_i = 0$. $AS_i = \sum_{j=i-N+1}^{i}(A_j)$. *If* $TYP_i < TYP_{i-1}$, $B_i = TYP_i * vol_i$; *else* $B_i = 0$. $BS_i = \sum_{j=i-N+1}^{i}(B_j)$. $V_i = AS_i / BS_i$, $MFI_i = 100 - (100/(1 + V_i))$. *Where* $N = 10$.

Factor32: $MI_i = close_i - close_{i-1}$, $AMI_i = SMA(MI_i, N, 1)$, $DIF_i = MA(AMI_{i-1}, n1) - MA(AMI_{i-1}, n2)$, $MICD_i = SMA(DIF_i, 5, 1)$, *where* $N = 10$, $n1 = 5$, $n2 = 10$.

Factor33: $MoneyFlow = \frac{(high_i + low_i + close_i)}{3} * vol_i$.

Factor34: $MTM_i = close_i - close_{i-N+1}$, $MTM\_MA = SMA(MTM_i, N, 1)$, *where* $N = 10$.

Factor35: If $close_i > close_{i-1}$, $vol1_i = vol_i$; *else* $vol1_i = 0$. *If* $close_i < close_{i-1}$, $vol2_i = -vol_i$; *else* $vol2_i = 0$. $OBV = \sum_{j=i-N+1}^{i}(vol1_j) + \sum_{j=i-N+1}^{i}(vol2_j)$, *where* $N = 10$.

Factor36: If $close_i > close_{i-1}$, $num_i = 1$, *else* $num_i = 0$. $PSY_i = \frac{\sum_{j=i-N+1}^{i}(num_j)}{N} * 100$. $PSY\_MA_i = MA(PSY_i, K)$, *where* $N = 10$, $K = 5$.

Factor37: If $vol_i > vol_{i-1}$, $PV_i = \frac{close_i}{close_{i-1}}$; *If* $vol_i \le vol_{i-1}$, $PV_i = 1$. $PVI_i = PVI_{i-1} * PV_i$, $MPVI_i = MA(PVI_i, N)$, *where* $PVI_1 = 100$, $N = 10$.

Factor38: $V_i = \frac{close_i - close_{i-1}}{close_{i-1}} * vol_i$, $PVT_i = \sum_{j=i-N+1}^{i}(V_j)$, *where* $N = 10$.

Factor39: $RC_i = \frac{close_i}{close_{i-N1+1}}$, $ARC_i = SMA(RC_{i-1}, N1, 1)$, $DIF_i = MA(ARC_{i-1}, N1) - MA(ARC_{i-1}, N2)$. $RCCD_i = SMA(DIF_i, N1, 1)$. *Where* $N1 = 5$, $N2 = 10$.

Factor40: $RC_i = \frac{close_i}{close_{i-N+1}}$, *where* $N = 10$.

Factor41: $ROC_i = \frac{(close_i - close_{i-N+1})}{close_{i-N+1}}$, $N = 10$.

Factor42: $A_i = max([close_i - close_{i-1}, 0])$, $B_i = abs(close_i - close_{i-1})$, $RSI_i = \frac{SMA(A_i, N, 1)}{SMA(B_i, N, 1)} * 100$, *where* $N = 10$.

Factor43: If $close_i > close_{i-1}$, $UP_i = \sqrt{\frac{\sum_{j=i-N+1}^{i}(close_j - mean([close_{i-N+1}, \cdots, close_i]))}{(N-1)}}$, *else* $UP_i = 0$. *If* $close_i < close_{i-1}$, $DOWN_i = \sqrt{\frac{\sum_{j=i-N+1}^{i}(close_j - mean([close_{i-N+1}, \cdots, close_i]))}{(N-1)}}$, *else* $DOWN_i = 0$. $AUP_i = SMA(UP_i, N, 1)$, $ADOWN_i = SMA(DOWN_i, N, 1)$, $RVI_i = \frac{AUP_i}{(AUP_i + ADOWN_i)} * 100$, *where* $N = 10$.

Factor44: If $close_i < close_{i-1}$, $SRMI = \frac{close_i - close_{i-1}}{close_{i-1}}$; *if* $close_i > close_{i-1}$, $SRMI = \frac{close_i - close_{i-1}}{close_i}$; *if* $close_i = close_{i-1}$, $SRMI = 0$.

Factor45: $A = max([close_i - close_{i-1}, 0])$, $B_i = abs(close_i - close_{i-1})$, $RSI_i = \frac{SMA(A_i, N, 1)}{SMA(B_i, N, 1)} * 100$, $StochRSI_i = \frac{RSI_i - min([RSI_{i-N+1}, \cdots, RSI_i])}{max([RSI_{i-N+1}, \cdots, RSI_i]) - min([RSI_{i-N+1}, \cdots, RSI_i])}$, *where* $N = 10$.

Factor46: $A_i = EMA(EMA(EMA(ln(close_i), n), n), n)$, $TRIX_i = \frac{A_i - A_{i-1}}{A_{i-1}}$, *where* $N = 10$.

Factor47: $R_i = \frac{close_i - min([close_{i-N+1}, \cdots, close_i])}{max([close_{i-N+1}, \cdots, close_i])} * 100$, $Ulcer_i = MA(R_i, N)$, *where* $N = 10$.

Factor48: $TH_i = max(high_i, close_{i-1})$, $TL_i = min(low_i, close_{i-1})$, $ACC1_i = \left(close_i - \sum_{j=i-N1+1}^{i}(TL_j)\right) / \sum_{j=i-N1+1}^{i}(TH_j - TL_j)$, $ACC2_i = \left(close_i - \sum_{j=i-N2+1}^{i}(TL_j)\right) / \sum_{j=i-N2+1}^{i}(TH_j - TL_j)$, $ACC3_i = \left(close_i - \sum_{j=i-N2+1}^{i}(TL_j)\right) / \sum_{j=i-N2+1}^{i}(TH_j - TL_j)$, $UOS_i = (ACC1_i * N2 * N3 + ACC2_i * N1 * N3 + ACC3_i * N1 * N2) * 100 / (N1 * N2 + N2 * N3 + N1 * N3)$, *where* $N1 = 3$, $N2 = 5$, $N3 = 10$.

Factor49: $std_i = \sqrt{\frac{\sum_{j=i-M+1}^{i}(close_j - mean([close_{i-M+1}, \cdots, close_i]))}{(M-1)}}$, std_h *is the highest value in dataset* $[std_{i-N+1}, \cdots, std_h, \cdots, std_i]$. $VR_i = N - (h - (i - N + 1))$, $AVR_i = EMA(VR_i, 3)$, *where* $M = 5$, $N = 10$.

Factor50: $VEMA_i = EMA(vol_i, N)$, $N = 10$.

Factor51: $DIFF_i = EMA(vol_i, N1) - EMA(vol_i, N2)$, $DEA_i = EMA(DIFF_i, N1)$, $VMACD_i = DIFF_i - DEA_i$, *where* $N1 = 5$, $N2 = 10$.

Factor52: $VOSC = \frac{(MA(vol_i, N1) - MA(vol_i, N2))}{MA(vol_i, N1)} * 100$, *where* $N1 = 5$, $N2 = 10$.

Factor53: $VROC = \frac{(vol_i - vol_{i-N+1})}{vol_{i-N+1}} * 100$, *where* $M = 5$.

Factor54: $VA_i = max([vol_i - vol_{i-1}, 0])$, $VB_i = abs(vol_i - vol_{i-1})$, $VRSI_i = \frac{SMA(VA_i, N, 1)}{SMA(VB_i, N, 1)} * 100$, *where* $N = 10$.

Factor55: If $close_i > close_{i-1}$, $RV_i = vol_i$; *else* $RV_i = 0$. *If* $close_i \le close_{i-1}$, $DV_i = vol_i$; *else* $DV_i = 0$. $AS_i = \sum_{j=i-N+1}^{i}(RV_j)$, $BS_i = \sum_{j=i-N+1}^{i}(DV_j)$. $VR_i = \frac{AS_i}{BS_i} * 100$, *where* $N = 10$.

References

1. K. P. Murphy, Machine learning: a probabilistic perspective, Chance, vol.27, no.2, pp.62-63, 2012.
2. K. J. Kim, I. Han, Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index, Expert System Appliance, vol.19, no.2, pp.125-132, 2000.
3. L. Nan, X. Liang, L. Xin, et al, Network environment and financial risk using machine learning and sentiment analysis, Human and Ecological Risk Assessment: An International Journal, vol.15, no.2, pp.26, 2009.
4. J. Patel, S. Shah, P. Thakkar, et al, Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques, Expert Systems with Applications An International Journal, vol.42, no.1, pp.259-268, 2015.

5. S. Yuan, W. Yingnian, Stock Trend Prediction: Based on Machine Learning Methods, UCLA Electronic Theses and Dissertations, 2018.
6. J. Patel, S. Shah, P. Thakkar, et al, Predicting stock market index using fusion of machine learning techniques, Expert Systems with Applications, vol.42, no.4, pp.2162-2172, 2015.
7. W. Huang, Y. Nakamor, S. Y. Wang, Forecasting stock market movement direction with support vector machine, Computers and Operations Research, vol.32, no.10, pp.2513-2522, 2005.
8. S. Sohangir, W. Ding, A. Pomeranets, et al, Big data: deep learning for financial sentiment analysis, Journal of Big Data, vol.5, no.1, pp.3, 2018.
9. M. Abe, H. Nakayama, Deep learning for forecasting stock returns in the cross-section, Pacific Asia Conference on Knowledge Discovery and Data Mining, 2018.
10. E. M. Attua, Using multiple linear regression techniques to quantify carbon stocks of fallow vegetation in the tropics, West African Journal of Applied Ecology, vol.12, no.1, 2009.
11. S. Hochreiter, J.Schmidhuber, Long short-term memory, Neural Computation, vol.9, no.8, pp.1735-1780, 1997.
12. J. A. K. Suykens, J. Vandewalle, Least squares support vector machine classifiers, Neural Processing Letters, vol.9, no.3, pp.293-300, 1999.
13. V. Svetnik, A. Liaw, et al, Random forest: a classification and regression tool for compound classification and qsar modeling, Journal of Chemical Information and Computer Sciences, vol.43, no.6, pp.1947, 2003.
14. S. R. Joelsson, J. A. Benediktsson, J. R. Sveinsson, Random forest classifiers for hyperspectral data, IEEE International Geoscience and Remote Sensing Symposium, 2005.
15. P. D. Allison, Logistic Regression Using the SAS System: Theory and Application, SAS Publishing, 1999.