A Case Study of Stock Investment Based on Data Mining Techniques

Muh-Cherng Wu[1*], Pei-Chun Liu[2], and Hui-Chih Hung[3]
Department of Industrial Engineering and Management, National Chiao Tung University, Taiwan
mcwu512@gmail.com[1], jiem712@gmail.com[2], hhc@nctu.edu.tw[3]
**Corresponding Author*

Received 25 February 2019; received in revised form 1 April 2019; accepted 2 May 2019

Abstract

Data mining techniques have been widely applied to predict stock prices. This study intends to predict whether a particular day is a "buy-day" or a "not-buy-day". A day is called a buy-day if the stock closing price is expected to rise over 10% in a coming period (say, 80 days); otherwise, it is called a not-buy-day. We applied to 12 data mining (DM) techniques to make the buy-day decision, which is essentially a binary classification problem. For each DM technique, 13 economic and stock trading variables are selected as input features and the output involves two states (buy-day or not-buy-day). The stock price of a Taiwanese company (TSMC) in 10 years (Jan. 2007 - Dec. 2016) is used as a case study. Data of the first 8 years are used in training and the last 2 years are used in testing. Numerical experiments reveal that average annual rate of return ranges from 15%-25% for the 12 DM techniques; and the one (k-Nearest Neighbors) outperforms the others.

Keywords: Binary classification, data mining, stock price prediction, stock investment

1. Introduction

Data mining (DM) techniques have been widely in predicting stock prices. Most prior studies focused on predicting stock prices/indices of next trading day. Considering 40 possible input variables, Laboissiere *et al*. (2015) forecasted the maximum and minimum stock prices of power distribution companies in the next day. Yan *et al*. (2017) considered six input variables and predicted the closing-index of Shanghai composite index in the next day. Zhang *et al*. (2017) proposed a hybrid data mining method to predict four stock indices in the next day. Instead of predicting stock prices, some studies (Inthachot *et al*. 2016; Liu *et al*. 2016; Chong *et al*. 2017) predicted the up/down of a stock in next day. Furthermore, Oliveira *et al*. (2017) used microblogs (e.g., Twitter) to forecast daily stock market variables like trading volume in next trading day.

A few studies attempted to predict stock prices in a future time horizon. For example, Zhang *et al*. (2016) proposed a method, which models the stock price trend (up, down, or flat) in a certain period of time (e.g., 30 days), and applied machine learning techniques to classify the period in order to identify buy/sell points. Shynkevich *et al*. (2017) examined the performance of a predictive system in different combinations of forecast horizon (1-30 days) and input window length (3-30 days), by using technical indicators as input and future stock price movements as output.

Considering a future time horizon (say, 80 days), this research attempts to predict whether a day is a "buy-day" or a "not-buy-day". A day is a buy-day if the stock closing price is expected to rise over 10% in the coming time horizon (e.g., 80 days). The "buy-day" decision is a binary classification problem, which herein is to be solved by applying 12 different data mining (DM) techniques. For each DM technique, 13 economic and stock trading variables are used as input variables (also called features); and the output involves two states (buy-day or not-buy-day). The stock prices of a Taiwanese company (TSMC) in 10 years (Jan. 2007 - Dec. 2016) are used as a case study. Given a DM technique, data of the first 8 years is used to train a classifier; and data of the last 2 years is used to test the classifier. Numerical experiments reveal that average annual rate of return ranges from 15%-25% for the 12 DM techniques; and the one (k-Nearest Neighbors) outperforms the others.

The remainder of this paper is organized as follows. Section 2 presents the 12 DM techniques and explains the development and evaluation of a classifier based on a DM technique. Section 3 describes the 13 input variables (features). Section 4 reveals numerical experiments. Conclusions are in Section 5.

2. Development and Evaluation of Classifiers

The 12 data mining techniques to be examined have been widely applied to solve various binary classification problems. Details of their algorithms can be accessed in dada mining tutorials (Witten et. al. 2005). These 12 DM techniques involve k-Nearest Neighbors, CART

(Classification and Regression Tree), Bagged CART, C5.0, Stochastic Gradient Boosting, Logistic Regression, Regularized Logistic Regression, Linear Discriminate Analysis, Random Forest, Neural Network, and Support Vector Machine.

Each DM technique can be used to develop a classifier based on its distinct algorithm. Development of a classifier is called training; and performance evaluation of the trained classifier is called testing. In the following, we use neural network (a particular DM technique) as an example to explain the core ideas of training (classifier development) and testing (classifier evaluation).

The architecture of a neural network (NN) model is shown in Figure 1, which involves three layers: input layer, hidden layer, and output layer. Each layer comprises a set of nodes. A link is built between a pair of node (i, j), where i denotes each node of a particular layer (e.g., input layer) and j denotes each node of its next layer (e.g., hidden layer). And a weighting parameter w_{ij} shall be defined on such a link. Development or training of a classifier is to appropriately determine all link weights (w_{ij}) based on the algorithm of a DM technique.

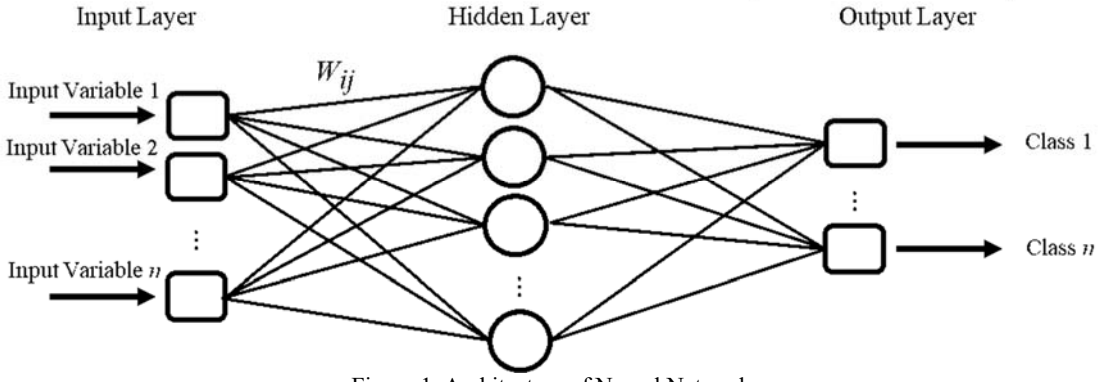

Figure 1: Architecture of Neural Network

The NN technique can be used to develop and evaluate a classifier based on a given data set $S = \{\bar{x}_t, \bar{y}_t | t=1,\ldots,n\}$, in which $\bar{x}_t$ represents an input vector and $\bar{y}_t$ represents the output vector at time t. The data set S is divided into two sets $S = S_1 \cup S_2$, where training set $S_1 = \{\bar{x}_t, \bar{y}_t | t = 1,\ldots,m\}$ and testing set $S_2 = \{\bar{x}_t, \bar{y}_t | t = m+1,\ldots,n\}$. Training set S_1 is used to train or develop an NN classifier by determining the weights of links (w_{ij}) so that a mapping function $\hat{\bar{y}}_t = f(\bar{x}_t)$ with a high classification accuracy is formed. Once the classifier has been trained (i.e., w_{ij} has been determined); testing set S_2 is used to evaluate the classification accuracy of the classifier.

Now we proceed to introduce four performance metrics in addressing a binary classification problem. See Table 1, the classification results can be of four categories: true positive (TP), false positive (FP), false negative (FN), and true negative (TN). To justify the capacity of a trained classifier, four performance metrics are typically examined, which are defined as follows:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

$$Specificity = \frac{TN}{TN+FP} \quad (2)$$

$$Sensitivity = \frac{TP}{TP+FN} \quad (3)$$

$$Precision = \frac{TP}{TP+FP} \quad (4)$$

Accuracy denotes the average classification capacity of the classifier, which is also used as the criteria in training the classifier. The three other performance metrics may not be mutually correlated; and are difficult to combine together to justify the aggregate performance of using such a classifier in stock investment. Therefore, by assuming a given trading strategy (called *one batch trading policy*), we compute annual rate of return (ARR) to evaluate the performance of a classifier.

Table 1: Definitions of TP, FP, TN, and TN.

		Original data	
		buy-day	not-buy-day
Prediction	buy-day	True Positive (TP)	False Positive (FP)
	not-buy-day	False Negative (FN)	True Negative (TN)

3. Input Features and Portfolios

To form an effective classifier, we have to select appropriate input variables (features), which are variables that have a substantial impact on the future trend of stock price. In this study, 13 input variables are examined, which are of two categories: macroeconomic category and stock trading category.

The macroeconomics category involves 3 variables (X_1, X_2, and X_3). The first two variables (X_1 and X_2) model the impacts of *money supply*; and variable X_3 denotes the TAIEX Index closing price; Money supply, an aggregate measure of money, includes currency and liquid

instruments (e.g., deposits in banks). Money supply can be defined in narrow or broad scope. According to the Central Bank of Taiwan, M1B is defined in narrow scope, which includes currency and some bank deposits that can be converted easily to cash. M2 is defined in broad scope, which includes M1B and certain bank deposits and funds that take a longer time to be converted to cash.

X_1 = *M1B annual growth rate*

$$= \frac{\text{(M1B in month i of year t) -(M1B in month i of year t-1)}}{\text{(M1B in month i of year t-1)}} \quad (5)$$

X_2 = *M1B annual growth rate–M2 annual growth rate* (6)

X_3 = *TAIEX Index closing price* (7)

The stock trading category involve 10 variables (X_4, …, X_{13}). Of these variables, four ones (X_4, …, X_7) are to model the stock closing prices of four consecutive days, in which X_4 = the closing price of day t; X_5 = the closing price of day t-1; X_6 = the closing price of day t-2; X_7 = the closing price of day t-3. Herein, day t denotes "today" and we attempt to justify if "tomorrow" (day t+1) is a "buy-day".

Variable $X_8 = u/k$ is to model the *psychological line* of the last k (herein we set k = 24) days, in which u denotes the number of days that are "up" against "yesterday" in terms of closing price. Variable $X_9 = (s_t - \bar{s}_{t,n})/\bar{s}_{t,n}$ (called *bias ratio*) is to mode the trend of stock price for the last n = 10 days, where s_t denotes the stock closing price of today (say, day t) and $\bar{s}_t$ denotes the average stock closing price of the last n days.

Variable X_{10} models the trading volume of day t; variable X_{11} models the monthly trading turnover rate (monthly trading shares / outstanding shares) at day t. Variable X_{12} = $(f_t - f_{t-1})$ is intended to model the potential momentum to buy stock, where f_t models the balance of financing for buying stocks at day t. Variable $X_{13} = (s_t - s_{t-1})$ is intended to model the potential momentum to sell the stock where s_t models the balance of securities loans at day t.

4. Numerical Experiments

A stock (TSMC) listed in Taiwan Stock Exchange is used to test the 12 DM techniques. TSMC Inc. is the largest company in Taiwan in terms of market capitalization. The training and testing of these classifiers are implemented by *R* programming language based on *neuralnet* package and carried out in the environment of Inter(R) Core(TM) CPU 3.10GHz and 16.0GB.

Data set for training and testing ranges from Jan. 2007 to Dec. 2016 (10 years); 80% of the data set is used for training and the remaining 20% is for testing. *Table 2* shows the distributions of "buy-day" and "not-buy-day" in the data sets. Notice that the percentage of "buy-day" and that of "not-buy-day" are both close to 50% (i.e., data sets are quite "balance").

The accuracy in training stage and the four performance metrics (accuracy, specificity, sensitivity and precision) in testing stage are shown in *Table 3*, which reveals that the accuracy of training stage is much better than that of testing stage. To evaluate performance of each DM classifier, we propose a trading strategy in order to compute the annual rate of return (ARR) in stock investment.

The trading strategy is called one batch trading policy. Suppose we have a bank account for stock investment, initially with a balance B0. Once a "predicted buy-day" appears, balance in the account is wholly devoted to buy the stock and the balance becomes zero. In the coming time horizon (80 days), whenever the stock rises over 10%, we sell all the stocks. If the stock price never rises over 10% in the coming time horizon, we shall sell all the stocks at opening price at the end day of the time horizon. Under the trading strategy, the annual rate of return (ARR) for each of the 12 DM techniques is shown in Table 4, which ranges from 15.0% to 25.5%. The DM technique k-Nearest Neighbor outperforms the others.

See Table 3 and Table 4, the neural network technique reveals a very high sensitivity (98.5%) and a very high precision (96.2%) in the testing stage, which substantially outperforms the other DM techniques. However, it ARR (annual rate of return) is only 19.0%, substantially lower than that (25.5%) of k-Nearest Neighbors. This is due that TSMC in the testing 2 years (Jan. 2015-Dec. 2016) is in a business growing stage. Its stock prices tend to have a rise trend in this period. Therefore, a "not-buy-day" may have a positive rate of return (i.e., positive but lower than 10%). This implies that a misclassification of "buy-day" might even impose a positive contribution on rate of return. This finding sheds a light for the need for developing a multiple-class classifier in order to highly correlate the predictive accuracy of a classifier to its annual rate of return.

Table 2: Percentage of Buy-Day and Not-Buy-Day.

Data Set	Percentage of Data Set	Buy-Day		Not-Buy-Day	
		Numbers	Percentage	Numbers	Percentage
Original Data Set	100%	1,236	51.62%	1,158	48.37%
Training Set	80%	955	49.87%	960	50.13%
Testing Set	20%	281	58.66%	198	41.34%

Table 3: Classification Performance Metrics in Training and Testing.

Algorithms	Training	Testing			
	Accuracy	Accuracy	Specificity	Sensitivity	Precision
k-Nearest Neighbors	90.0%	56.2%	80.1%	22.2%	59.4%
Bagged CART	95.2%	66.0%	66.9%	64.7%	72.9%
Stochastic Gradient Boosting	91.6%	64.1%	86.1%	32.8%	64.5%
Naïve Bayes	74.4%	52.4%	71.2%	25.8%	57.6%
Logistic Regression	70.7%	59.7%	98.6%	4.6%	59.4%
Regularized Logistic Regression	70.8%	59.3%	98.9%	3.0%	59.2%
Classification and Regression Trees	74.7%	59.3%	81.9%	27.3%	61.5%
Linear Discriminate Analysis	70.8%	58.7%	99.3%	1.0%	58.7%
Random Forest	96.0%	63.5%	69.8%	54.6%	68.5%
Neural Network	97.3%	56.4%	26.7%	98.5%	96.2%
C5.0	96.5%	48.0%	11.7%	99.5%	97.1%
Support Vector Machine	81.0%	44.5%	42.4%	47.5%	53.4%

Table 4: Annual Rate of Return and Number of Investments.

Data Mining Algorithms	Annual Rate of Return				Number of Investments (over 10% return rate)	Number of Investments (less than 10% return rate)	Total Number of Investments
	1st year	2nd year	Sum	Average			
k-Nearest Neighbors	11.5%	39.5%	50.9%	25.5%	5	3	8
Bagged CART	9.2%	37.3%	46.5%	23.3%	5	2	7
Stochastic Gradient Boosting	14.2%	31.3%	45.5%	22.8%	5	3	8
Naïve Bayes	7.5%	7.5%	43.9%	21.9%	4	4	8
Logistic Regression	11.5%	28.6%	40.0%	20.0%	3	5	8
Regularized Logistic Regression	11.5%	28.6%	40.0%	20.0%	3	5	8
Classification and Regression Trees	11.5%	28.6%	40.0%	20.0%	3	5	8
Linear Discriminate Analysis	11.5%	28.6%	40.0%	20.0%	3	5	8
Random Forest	9.2%	29.8%	39.0%	19.5%	4	3	7
Neural Network	20.2%	17.9%	38.1%	19.0%	3	1	4
C5.0	20.2%	10.1%	30.3%	15.2%	4	0	4
Support Vector Machine	13.2%	16.7%	29.9%	15.0%	3	3	6

5. Conclusions

This paper presents the application of 12 data mining techniques to predict the "buy-day" for stock investment. Data mining techniques have been widely used in predicting stock prices; yet most studies focused on the prediction of next trading day. This research is distinguished in predicting whether a trading day is a "buy-day" by considering a future time horizon (say, 80 days), where a trading day is a "buy-day" if the stock is expected to rise over 10% in the coming 80 days.

We address 13 macroeconomic and stock trading variables as input features of a classifier to be developed based on a DM technique. These input variables include the modeling of money supply, Taiwan stock exchange index, stock daily prices for 4 consecutive days, stock trends for the last 10 and 24 days, stock trading volume and turnover rate, and various momentums to buy or sell.

Numerical experiments reveal that the average ARR (annual rate of return) ranges from 15.0% to 25.5% for the 12 DM techniques. These experiments imply that the examined input features have a substantial impact on the rate of return. One extension of this research is the application of this approach with possible enhancements of input variables to other stocks. The other extension is the development of a multiple-class classifier in order to highly correlate the classifier prediction accuracy to its annual rate of return.

References

Chong, E., Han, C., & Park, F. C. (2017). Deep learning networks for stock market analysis and prediction: Methodology, data representations, and case studies. *Expert Systems with Applications*, *83*, 187-205.

Inthachot, M., Boonjing, V., & Intakosum, S. (2016). Artificial neural network and genetic algorithm hybrid intelligence for predicting thai stock price index trend. *Computational Intelligence and Neuroscience*.

Laboissiere, L. A., Fernandes, R. A., & Lage, G. G. (2015). Maximum and minimum stock price forecasting of brazilian power distribution companies based on artificial neural networks. *Applied Soft Computing*, *35*, 66-74.

Liu, C., Wang, J., Xiao, D., & Liang, Q. (2016). Forecasting S&P 500 stock index using statistical learning models. *Open Journal of Statistics*, *6*, 1067.

Oliveira, N., Cortez, P., & Areal, N. (2017). The impact of microblogging data for stock market prediction: using twitter to predict returns, volatility, trading volume and survey sentiment indices. *Expert Systems with Applications*, *73*, 125-144.

Shynkevich, Y., McGinnity, T. M., Coleman, S. A., Belatreche, A., & Li, Y. (2017). Forecasting price movements using technical indicators: Investigating the impact of varying input window length. *Neurocomputing*, *264*, 71-88.

Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2005). *Data Mining: Practical Machine Learning Tools and Techniques*. (2nd ed.). Elsevier.

Yan, D., Zhou, Q., Wang, J., & Zhang, N. (2017). Bayesian regularisation neural network based on artificial intelligence optimisation. *International Journal of Production Research*, *55*, 2266-2287.

Zhang, N., Lin, A., & Shang, P. (2017). Multidimensional k-nearest neighbor model based on EEMD for financial time series forecasting. *Physica A: Statistical Mechanics and its Applications*, *477*, 161-173.

Zhang, X. D., Li, A., & Pan, R. (2016). Stock trend prediction based on a new status box method and AdaBoost probabilistic support vector machine. *Applied Soft Computing*, *49*, 385-398.

About Authors

Muh-Cherng Wu is a professor at National Chiao Tung University, Taiwan. His recent research interests include the applications of data mining techniques and meta-heuristic algorithms.

Pei-Chun Liu is an engineer. She got her master degree in Industrial Engineering from National Chiao University.

Hui-Chih Hung is an associate professor at National Chiao Tung University, Taiwan. His recent research interests include scheduling and the applications of data mining techniques.