Finance, Markets and Valuation

Forecasting stock market trend: a comparison of machine learning algorithms

La predicción de la tendencia en los mercados bursátiles: una comparativa entre algoritmos de machine learning

Roberto Cervelló-Royo [1], **Francisco Guijarro** [2]

[1] *Economics and Social Sciences Department, Universitat Politècnica de València. Valencia, España. Email: rocerro@esp.upv.es*
[2] *Research Institute for Pure and Applied Mathematics, Universitat Politècnica de València. Valencia, España. Email: fraguima@upvnet.upv.es*

JEL: G11; G12; G17

Abstract

Forecasting the direction of stocks markets has become a popular research topic in recent years. Different approaches have been applied by researchers to address the prediction of market trends by considering technical indicators and chart patterns from technical analysis. This paper compares the performance of four machine learning algorithms to validate the forecasting ability of popular technical indicators in the technological NASDAQ index. Since the mathematical formulas used in the calculation of technical indicators comprise historical prices they will be related to the past trend of the market. We assume that forecasting performance increases when the trend is computed on a longer time horizon. Our results suggest that the random forest outperforms the other machine learning algorithms considered in our research, being able to forecast the 10-days ahead market trend, with an average accuracy of 80%.

Keywords: Trend forecasting; Stock markets; Random Forest; Deep Learning.

Resumen

La predicción de la tendencia en los mercados financieros se ha convertido en un área de investigación popular durante los últimos años. Los investigadores han aplicado diferentes enfoques metodológicos para abordar la predicción de la tendencia a través de indicadores técnicos y patrones chartistas del análisis técnico. Este trabajo compara el rendimiento de cuatro algoritmos de machine learning para validar la habilidad predictiva sobre la tendencia del índice tecnológico NASDAQ de algunos indicadores técnicos habitualmente utilizados entre los traders. Debido a que los indicadores técnicos se obtienen a partir de fórmulas matemáticas aplicadas sobre los precios históricos, asumimos que su valor está ligado a la tendencia pasada de los mercados. Nuestra hipótesis es que la capacidad predictiva de estos modelos aumenta conforme se amplía el horizonte temporal del análisis.Nuestros resultados sugieren que el algoritmo de random forest supera al resto de algoritmos analizados en nuestro trabajo, siendo capaz de predecir la tendencia de los próximos 10 días con una fiabilidad promedio del 80 %.

Keywords: Predicción de la tendencia; Mercados bursátiles; Random Forest; Deep Learning.

DOI:
10.46503/NLUF8557

Corresponding author
Francisco Guijarro

Received: 25 Mar 2020
Revised: 26 Apr 2020
Accepted: 12 May 2020

Finance, Markets and Valuation
ISSN 2530-3163.

1 Introduction

The use of machine learning techniques allows exploring new opportunities and developments for investment in the field of financial markets (Efendi, Arbaiy, y Deris, 2018; Krauss, Do, y Huck, 2017). In this way, technological developments provide private investors access to any financial market, so that they can execute their buy and sell decisions in real time by using tools for price analysis and future trend prediction (Bekiros, 2010; Kara, Boyacioglu, y Ömer Kaan Baykan, 2011; Kim, 2003; Kim y Han, 2000; Leung, Daouk, y Chen, 2000).

In the last decade, the way of investing has changed at least from two perspectives. First, investors have access to financial markets in real time. The use and proliferation of new technologies, the speed of access to data and its quantitative treatment, allow them to make investment decisions quickly and execute them through cross-platform trading software. Second, artificial intelligence, supported precisely by technological advances, has led to a new way of investing where decisions can be carried out by handling huge amounts of information, and quantifying the risks associated with them much more precisely (Arévalo, García, Guijarro, y Peris, 2017; Cervelló-Royo, Guijarro, y Michniuk, 2015; Chen y Chen, 2016; Lee, 2009). The increasing global access to large databases with prices and technical indicators obtained from it, allows investors to make their decisions regardless of the emotional component, focusing on a more rational framework based on the analysis of the available quantitative information, and where other sources of information, like market sentiment, can also be added to price information (Nguyen, Shirai, y Velcin, 2015).

In this context, machine learning techniques can make an outstanding contribution, facilitating rational decision making and enabling new opportunities for investors and entrepreneurs (Abbas, 2018).

Markets and individual stocks move in trends, and history and economic cycles tend to repeat themselves (Kirkpatrick II y Dahlquist, 2010). There is abundant literature where researchers use historical prices in order to predict the stock price behavior (Ballings, den Poel, Hespeels, y Gryp, 2015; García, Guijarro, Oliver, y Tamošiūnienė, 2018) and discover patterns in data by means of machine learning techniques (Cervelló-Royo y cols., 2015; Göçken, Özçalıcı, Boru, y Dosdoğru, 2016; Li, Li, y Li, 2017; Patel, Shah, Thakkar, y Kotecha, 2015a, 2015b).

Most current trading platforms incorporate tools for stock market analysis of traded financial assets, from individual stocks to indices, financial futures, commodities, currencies (Forex market) or, more recently, cryptocurrencies. New tools include indicators and technical oscillators, which are mathematical calculations obtained from the historical price and / or volume of the financial asset and that are used by investors to try to predict the future behaviour of stock prices. The number of indicators is large, and in many cases the signals offered by some may contradict those offered by others. Analysis carried out should help identify investment opportunities, and investors should be able to handle a large amount of information in real time (García, González-Bueno, Guijarro, y Oliver, 2020). In this context, machine learning methods can provide solutions that the individual investor cannot handle efficiently, by simultaneously analyzing different indicators and their interactions, and thus predicting market behavior (Ballings y cols., 2015; Huang, Yang, y Chuang, 2008; Ni, Ni, y Gao, 2011; Weng, Lu, Wang, Megahed, y Martinez, 2018; Zhang, Cui, Xu, Li, y Li, 2018) from the historical analysis of a large database in minimum time. Machine learning techniques can be applied both to forecast stock prices and trends (Bekiros, 2010; García y cols., 2018; Hu, Tang, Zhang, y Wang, 2018; Kara y cols., 2011; Kim, 2003; Kim y Han, 2000; Leung y cols., 2000). This latter one will be the object of this study. That is, we will not forecast future closing prices but the market trend for different time

horizons. For this purpose, we will focus on stocks from the NASDAQ 100 which lists the largest technological companies (Industrial, Technology, Retail, Telecommunications, Biotechnology, Healthcare, Transportation, Media and Service companies). Furthermore, companies listed on this index like Amazon, Apple or Facebook, are strong references for entrepreneurial ventures.

The paper is structured as follows: Section 2 describes the theoretical background of the applied learning machine methods and our forecasting models. Section 3 reports the empirical results from the application of the machine learning methods introduced in the previous section over a sample of 89 technological companies from NASDAQ-100 Index during the period from 30th March 2009 to 27th December 2019. Finally, our main conclusions are given in Section 4.

2 Machine learning methods for stock market trend forecasting

This Section briefly describes the methods used in our research for the classification forecasting of stock market trend. Our interest is focused on two possible states: bullish trend vs. bearish trend.

2.1 Random Forest

Random Forest is a decision tree algorithm, and it works as an ensemble algorithm in which more than one model is made (Breiman, 2001). It is effective on a wide range of data sets, since it needs relatively few parameters. Random forest allows to avoid overfitting and to have lots of trees. New data can be given to each of those trees and ask each and one of them for their prediction. When dealing with a classification problem the most popular answer can be chosen, and when focusing on a regression problem the mean of each tree's answer can be taken. It is random because during the training process all the training data is not given to each tree; some rows and / or columns are randomly hold back. All this makes each individual tree a bit less effective, but when their results are averaged together the whole is more accurate than any single one.

2.2 Deep Learning

Unlike the artificial neural networks of the past, modern Deep Learning (DL) provides training stability, generalization, and scalability with big data. According to Candel, Parmar, LeDell, y Arora (2016), "since it performs quite well in a number of diverse problems, Deep Learning is quickly becoming the algorithm of choice for the highest predictive accuracy".

Many layers of interconnected neuron units constitute the multi-layer-feed-forward neural networks. First, there will be an input layer to match the feature space, followed by multiple layers of nonlinearity, and ending with a linear regression or classification layer which should match the output space. Each non-output layer of the network will include bias units. The output of the entire network will be determined by the weights obtained by linking neurons and biases with other neurons (Ballings y cols., 2015). Thus, these weights are adapted to minimize the error on the labelled training data linking neurons and biases with other neurons and fully determining the output of the entire network; then and when these weights are adapted to minimize the error on the labelled training data, learning will occur. The main objective is to minimize a loss function, for each and one training example. Deep Learning architectures are models of hierarchical feature extraction, which usually involve multiple levels of nonlinearity. These Deep Learning models are able to learn useful representations of raw data and have shown high performance on complex data.

2.3 Gradient Boosting Machines

Gradient Boosting Machines (GBM) is a powerful machine learning algorithm that is part of the boosting family algorithms. GBM models have proved to have strong predictive performance and high flexibility in a wide range of data driven applications (Oppel y cols., 2012; Sayegh, Tate, y Ropkins, 2016; Touzani, Granderson, y Fernandes, 2018). Boosting algorithms were originally introduced for classification problems and like the random forest, it is a decision tree algorithm. The main goal is to combine iteratively several simple models, called "weak learners", to obtain a "strong learner" which allows improving the prediction accuracy. Friedman, Hastie, y Tibshirani (2000) extended the boosting to the regression by introducing the gradient boosting machines method. The GBM method constitutes a numerical optimization algorithm that looks for an additive model that minimizes the loss function.

Thus, using a training set $\{(x_1, y_1), \ldots, (x_n, y_n)\}$ of known values of x and corresponding values of y, the goal is to find an approximation $\hat{F}(x)$ to a function $F(x)$ that minimizes the expected value of this specified loss function $L(y, F(x))$:

$$\hat{F} = \underset{F}{\operatorname{argmin}} \mathbb{E}_{x,y} [L(y, F(x))] \tag{1}$$

The gradient boosting method assumes a real-valued y and seeks an approximation $\hat{F}(x)$ in the form of a weighted sum of functions $hi(x)$ from some class H, the so called base (or weak) learners:

$$\hat{F}(x) = \sum_{i=1}^{M} \gamma_i h_i(x) + \alpha \tag{2}$$

where α represents a constant. Thus, the GBM algorithm iteratively adds at each step a new decision tree which best reduces the loss function. In other words, the method tries to find an approximation $\hat{F}(x)$ that minimizes the average value of the loss function on the training set. It does so by starting with a model, consisting of a constant function $F_0(x)$), and incrementally expands it:

$$F_0(x) = \underset{\gamma}{\operatorname{argmin}} \sum_{i=1}^{n} L(y_i, \gamma) \tag{3}$$

In other words, when applying regression, the algorithm starts by initializing the model by a first guess, which is usually a decision tree that maximally reduces the loss function, at each step a new decision tree is fitted to the current residual and added to the previous model to update the residual.

Then for $k = 1$ to K (number of iterations) the pseudo-residuals should be computed and following one-dimensional optimization problem the model can be updated to:

$$F_k(x) = F_{k-1}(x) + \gamma_k h_k(x) \tag{4}$$

Then, the user provides a maximum number of iterations, and the algorithm continues to iterate until this maximum number of iterations is reached. An important part of the gradient boosting method is the regularization by means of shrinkage which consists in modifying the

update rule in the following way:

$$F_k(x) = F_{k-1}(x) + v\gamma_k h_k(x), \quad 0 < v \leq 1 \tag{5}$$

The learning rate v is a shrinkage parameter that allows the GBM algorithm to have better results if at each iterative step the contribution of the added decision tree is shrunk using this learning rate. The learning rate ranges from 0 to 1 and the smaller it is, the more accurate the model is. Nevertheless, a stronger shrinkage (smaller learning rate) implies a higher number of iterations to achieve convergence, because the value of the learning rate is inversely proportional to the number of iterations.

A representation of the algorithm can be simply represented by the following pseudo-code:

1. First, we should select the depth of the decision trees d, the number of iterations K, the learning rate v, and the subsample fraction η.
2. Initialization: set the residual $r_0 = y$ and $f = 0$. The mean value of y has also been suggested as an initial guess of f (Liaw y Wiener, 2002).
3. For $k = 1, 2, \ldots, K$, do the following:
   a) Randomly choose a subsample $\{y_i, x_i\}^{N'}$ from the full training dataset, with N' is the number of data points corresponding to the fraction η
   b) Using $\{y_i, x_i\}^{N'}$ fit a decision tree f^k of depth d to the residual r_{k-1}
   c) Update f by adding the decision tree to the model $f(x) \leftarrow f(x) + vf^k(x)$
   d) Update the residuals $r_k \leftarrow r_{k-1} - vf^k(x)$

2.4 Generalized Linear Models

Generalized Linear Models (GLM) will allow estimating regression models for outcomes following exponential distributions. In addition to the Gaussian distribution, these models include Poisson, Binomial, Quasibinomial, Gamma, Ordinal, Negative Binomial and Tweedle distributions and/or regressions, as well as Binomial and Multinomial classifications. Each one serves for a different purpose, and depending on the kind, can be used either for prediction and/or classification, i.e. when considering as outcome a binary variable we are talking about logistic regression.

Thus, for binary classification problems where the response is a categorical variable with two levels, Logistic regression will be used. Given the data, Logistic regression will provide the probability of an observation belonging to an output category:

$$P_i = P_r\,(y = 1|x) \tag{6}$$

The probability distribution function can be written as:

$$P_i = P_r\,(y = 1|x_1, x_2, \ldots, x_k) = F\,(\alpha + \beta X_i) = F(Z_i) \tag{7}$$

Where P_i is the probability of the dependent variable being 1 subject to the explanatory variables. There are two basic types of binary choice models: the Logit model and the Probit model. In the Logit model, the functional form of $F(.)$ is the cumulative distribution function of the logarithmic distribution, while in the Probit model, the functional form of $F(.)$ is the

cumulative distribution function of a standard normal distribution. The Logit function is the most common function for the binomial family. Its inverse is the logistic function, which takes any real number and projects it onto the [0, 1] range as desired to model the probability of belonging to a class. We will use the Logit model, which is defined as follows:

$$P_i = P_r\left(y = 1|x_1, x_2, \ldots, x_k\right) = F(Z_i) = \frac{1}{1 + exp(-Z_i)} \tag{8}$$

and it can also be converted to

$$log\left(\frac{P_i}{1 - P_i}\right) = Z_i = \alpha + \beta X_i \tag{9}$$

3 Results

This Section presents the application of the machine learning methods introduced in the previous Section to a large database composed by technological firms from the NASDAQ-100 Index. We have compiled the open, high, low and close prices (OHLC) on a daily basis along with volume for those stocks listed in the NASDAQ-100 Index during the period 30th March 2009 to 27th December 2019. We excluded those stocks with missing values during the analysed period; hence, the database is eventually composed by 89 stocks. We have computed the trend indicator $I_{t,i}$ by comparing closing prices at times t (C_t) and $t - i$ (C_{t-i}). Equation 3 establishes a value of 1 in case the trend is positive, and 0 otherwise:

$$I_{t,i} = \begin{cases} 1 & C_t > C_{t-1} \\ 0 & C_t \leq C_{t-1} \end{cases} \tag{10}$$

This way, a value of 1 in the indicator denotes a bullish trend, while a value of 0 translates into a bearish trend. To extend the analysis, we have computed the trend by considering different distances between closing prices. The shortest version of the trend indicator used in this research is for $i = 1$, which establishes the difference between two consecutive closing prices. We have computed the trend indicators for different lagged values of the closing price: $i \in \{1, 5, 10\}$.

Price and volume information was extended with the technical indicators showed in Table **??**. The parameter values indicated in the table are the default values included in the TTR library of R software, as these are the commonly used in most trading platforms.

Table 1. Technical indicators and parameters used in the empirical research

Name of indicators and parameters	Parameter values
Aroon indicator	Number of periods to use in the calculation: 20
Average True Range (ATR)	Number of periods for moving average: 14
Bollinger Bands (BB)	Number of periods for moving average: 20 Number of standard deviations used (sd): 2
Chaikin Volatility	Number of periods for moving average: 10

Table 1. Technical indicators and parameters used in the empirical research

Name of indicators and parameters	Parameter values
Close-to-close volatility (volat_close)	Number of periods for the volatility estimate: 10 Number of periods per year: 260
Commodity Channel Index (CCI)	Number of periods for moving average: 20
Directional Movement Index (ADX)	Number of periods to use for direction calculation: 14
Exponential Moving Average (EMA)	Number of periods for moving average. we have used two versions of this technical indicator: 10 and 25
Money Flow Index (MFI)	Number of periods to use in volume calculations: 14
Moving Average Convergence / Divergence (MACD)	Number of periods for fast moving average: 12 Number of periods for slow moving average: 26 Number of periods for signal moving average: 9
On Balance Volume (OBV)	No parameters are needed, just price and volume
Parkinson formula for historical volatility (volat_parkinson)	Number of periods for the volatility estimate: 10 Number of periods per year: 260
Relative Strength Index (RSI)	Number of periods for moving average: 14
Stochastic Momentum Index (SMI)	Number of periods for initial smoothing: 2 Number of periods for double smoothing: 25 Number of periods for signal line: 9

Figure 1 shows the correlation matrix between all variables considered in the prediction of the market trend. We can observe that OHLC prices are highly correlated with technical indicators EMA10, EMA25, MACD and ATR. However, most of them show a low correlation coefficient among them. This is of strong interest for the purpose of our research; thus, we can conclude that indicators can contribute in a different and specific way to the explanation of trend behaviour.

The methods introduced in the previous section have been used to estimate the trend indicator, $\hat{I}^t$, by considering two possible scenarios. The first one only considers OHLC prices and volume from period $t-i$ in the prediction of trend indicator of period t; i.e., five variables in the forecasting of the trend. The second one extends these input variables with the 14 technical indicators referred in Table 1. According to previous literature (Kara y cols., 2011; Patel y cols., 2015b), we assume that technical indicators do have incremental value in the prediction of the market trend. OHLC prices and volume from $t-i$ just consider information preceding the closing price from period $t-i$. However, technical indicators are calculated by considering a larger period or prices and volumes, hence its potential ability to summarize the past behaviour of stocks. The different values considered for computing the lagged closing prices in Equation 3 allow to analyse the forecasting ability of technical indicators on the trend indicator. We assume that the larger the value of i, the more significant forecasting ability of technical indicators, as its calculation involves the use of historical prices and volumes.

We have extended the analysis with two additional scenarios. In the first one we have applied the machine learning methods to each individual stock. One limitation of this approach

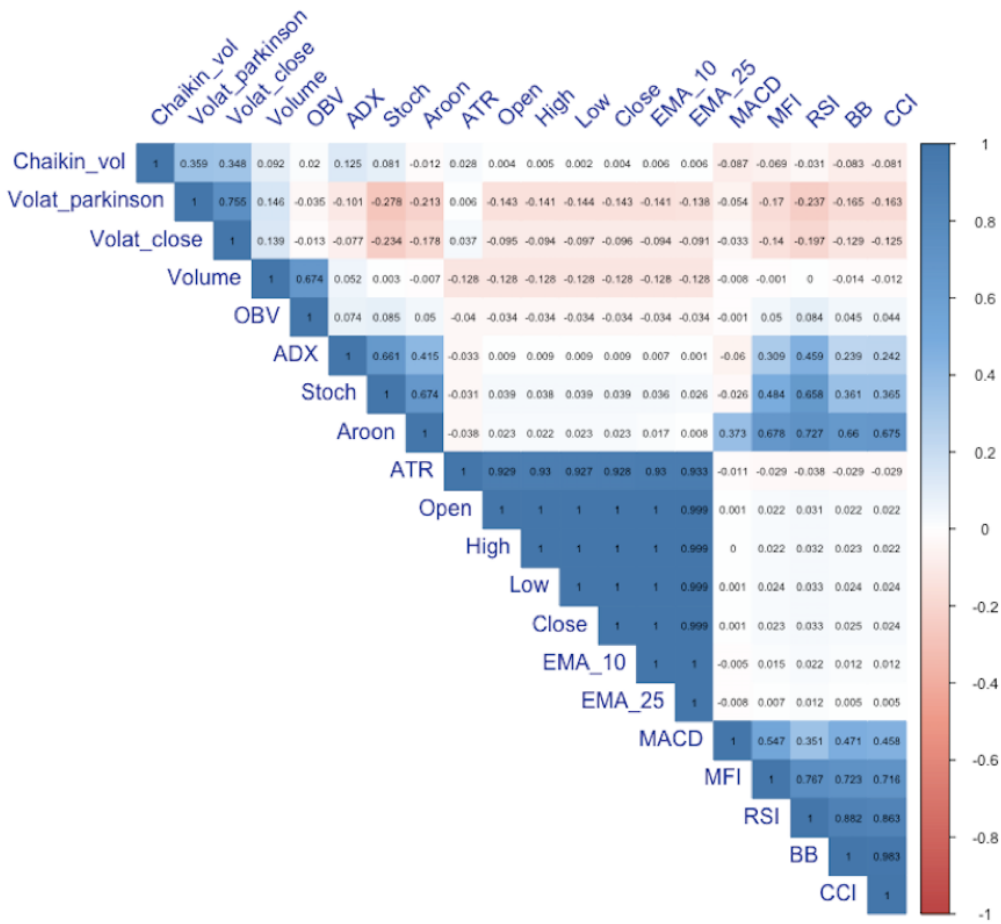

Figure 1. Correlation matrix for variables considered in the prediction of trend market

is that the sample size is constrained to the period analyzed: 2,516 daily observations per stock. This would potentially limit the effectiveness of the machine learning methods which are supposed to provide more accurate results with large datasets, therefore patterns can be more effectively captured. However, technical analysis assumes that price patterns are globally valid regardless of the considered stock (Cervelló-Royo y cols., 2015; Tsinaslanidis, 2018). This led us to establish a second scenario by considering all the stocks as a whole dataset, thus enlarging the original sample and gathering 223,924 observations in the final sample.

Table 2 summarizes the different scenarios considered in this research. The three versions of the trend indicator according to the lagging of closing prices ($i \in \{1, 5, 10\}$) are used for all scenarios.

We have used a common framework for the different machine learning techniques used in the research. The training set was randomly selected by considering the 80% of the sample while the remaining 20% was separated for testing. We carried out 50 experiments per each scenario and machine learning method to assess the consistency of the results. Table 3 shows the parameter values used in the different machine learning algorithms. These were taken as the default values in the h2o library from R Software.

Table 4 shows the average accuracy ratios obtained along the experiments carried out in our research. We have trained and tested 4 different machine learning methods: GBM, RF, DL

	Without technical indicators	With technical indicators
Individual stocks	Scenario A	Scenario B
	2,516 observations	2,516 observations
	5 trend predictors	19 trend predictors
Aggregated sample of stocks	Scenario C	Scenario D
	223,924 observations	223,924 observations
	5 trend predictors	19 trend predictors

Table 2. Simple size and number of predictors for the different scenarios considered in the research

Machine learning algorithms	Parameters	Value
Deep Learning	Number of folds for crossvalidation	5
	Hidden layers	(200, 200)
	epochs (number of iterations)	100
	rho (adaptive learning rate time decay factor)	0.99
	epsilon (adaptive learning rate smoothing factor)	1E-08
	Learning rate	0.005
Random forest	Number of folds for crossvalidation	5
	Number of trees	50
	Maximum tree depth	20
Gradient Boosting Machine	Number of folds for crossvalidation	5
	Number of trees	50
	Maximum tree depth	20
	Learning rate	0.1
Generalized Lineal Model	Number of folds for crossvalidation	5
	Family	Binomial
	Theta	1E-10

Table 3. Parameter values considered in the machine learning algorithms

and GLM. For all the considered scenarios, the table reports the average accuracy ratio for the 50 experiments along with the standard deviation of these values. The three alternatives of time horizon of the trend indicator are also reported. We can observe that the machine learning algorithms did not obtain satisfactory results in scenarios A, B and C. Despite the average accuracy ratios are above 50%, we have to consider that during the period analyzed the NASDAQ market has experienced a long bullish market. This translates into more bullish than bearish trading days, and thus the trend indicator $I_{t,i}$ has a biased mean value, greater than 0.5. We can observe that 52.3% of days were bullish (last column of Table 4). And this percentage is 55.1% for 5-days periods and 57.0% for 10-days periods. Therefore, a naïve forecasting model could obtain these values by unfailingly predicting the value 1 regardless of the inputs considered in the algorithm.

However, some results of scenario D are above these mean trend indicator values. In the case of $i = 1$, the percentage of bullish trading days is the same than the one obtained by all of the 4 machine learning algorithms: 52.3%. Interestingly, this accuracy ratio improves as the value of i increases. For $i = 5$, the percentage of bullish trading days is 55.1%, but all algorithms report a better accuracy ratio: 57.5% (GBM), 72.4% (RF), 56.2% (DL) and 56.0% (GLM). And the RF algorithm is still better when considering the $i = 10$ case. The percentage of bullish periods is 57.0%, while the RF obtains an average accuracy ratio of 80.8% and a standard deviation of 0.005. Despite GBM, DL and GLM get accuracy ratios above the 57.0% threshold, the successful ratio of RF is by large much better than the ones obtained by its competitors. The low values reported for the standard deviation of the accuracy ratios demonstrate the consistency of the algorithms in the computation of the market trend.

ML alg.	GBM	RF	DL	GLM	% $I_{t,i} = 1$
i			Scenario A		
1	0.533 (0.020)	0.532 (0.020)	0.533 (0.020)	0.533 (0.021)	0.523
5	0.520 (0.018)	0.520 (0.018)	0.519 (0.018)	0.521 (0.018)	0.551
10	0.523 (0.018)	0.522 (0.019)	0.523 (0.020)	0.522 (0.020)	0.570
i			Scenario B		
1	0.529 (0.022)	0.529 (0.023)	0.530 (0.022)	0.529 (0.023)	0.523
5	0.522 (0.020)	0.522 (0.020)	0.522 (0.019)	0.522 (0.020)	0.551
10	0.523 (0.022)	0.522 (0.022)	0.523 (0.023)	0.524 (0.022)	0.570
i			Scenario C		
1	0.523 (0.000)	0.523 (0.000)	0.523 (0.000)	0.523 (0.000)	0.523
5	0.550 (0.000)	0.550 (0.000)	0.550 (0.000)	0.550 (0.000)	0.551
10	0.577 (0.006)	0.582 (0.009)	0.574 (0.005)	0.574 (0.005)	0.570
i			Scenario D		
1	0.523 (0.004)	0.523 (0.004)	0.523 (0.004)	0.523 (0.004)	0.523
5	0.575 (0.005)	0.724 (0.008)	0.562 (0.004)	0.560 (0.004)	0.551
10	0.606 (0.007)	0.808 (0.005)	0.582 (0.006)	0.577 (0.004)	0.570

Table 4. Mean accuracy values obtained by the machine learning methods

Note: Values represent the mean accuracy obtained by each machine learning method in the different scenarios. Values in parenthesis are the standard deviation of the accuracy values. The experiments were carried out for 3 different lagged closing prices in the trend indicator ($i \in \{1, 5, 10\}$). Last column represents the percentage of bullish trend indicator.

4 Conclusions

Machine learning techniques have become of special interest in financial operations. In this vein, new technologies offer new business opportunities for investors in the financial markets.

This paper deals with the application of machine learning techniques in a well-known technological index like the NASDAQ 100. Thus, our main goal is the forecasting of market trends by comparing the performance of well-established machine learning algorithms. Technical analysis assumes that stock markets tend to repeat themselves, thus its main objective is to accurately identify patterns through different technical indicators. Our experiments are focused to assess the prediction ability of these indicators through the use of machine learning algorithms. The mathematical formulas used to calculate technical indicators include past price and volume, which summarizes the past behaviour of the market. Literature suggests that this can help to predict the future trend. Our results provide empirical evidence that technical indicators have poor predictive power for the short-term –the following trading day trend–. However, results significantly improve when the time horizon is expanded to forecast the 10 trading day trend. Random forest results in the best performance algorithm with an average accuracy ratio of 80.8%.

References

Abbas, S. A. (2018). Entrepreneurship and information technology businesses in economic crisis. *Entrepreneurship and Sustainability Issues*, *5*(3), 682–692. doi: https://doi.org/10.9770/jesi.2018.5.3(20)

Arévalo, R., García, J., Guijarro, F., y Peris, A. (2017). A dynamic trading rule based on filtered flag pattern recognition for stock market price forecasting. *Expert Systems with Applications*, *81*, 177–192. doi: https://doi.org/10.1016/j.eswa.2017.03.028

Ballings, M., den Poel, D. V., Hespeels, N., y Gryp, R. (2015). Evaluating multiple classifiers for stock price direction prediction. *Expert Systems with Applications*, *42*(20), 7046–7056. doi: https://doi.org/10.1016/j.eswa.2015.05.013

Bekiros, S. D. (2010). Fuzzy adaptive decision-making for boundedly rational traders in speculative stock markets. *European Journal of Operational Research*, *202*(1), 285–293. doi: https://doi.org/10.1016/j.ejor.2009.04.015

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32.

Candel, A., Parmar, V., LeDell, E., y Arora, A. (2016). Deep learning with H2O. *H2O. AI Inc*.

Cervelló-Royo, R., Guijarro, F., y Michniuk, K. (2015). Stock market trading rule based on pattern recognition and technical analysis: Forecasting the DJIA index with intraday data. *Expert Systems with Applications*, *42*(14), 5963–5975. doi: https://doi.org/10.1016/j.eswa.2015.03.017

Chen, T.-l., y Chen, F.-y. (2016). An intelligent pattern recognition model for supporting investment decisions in stock market. *Information Sciences*, *346-347*, 261–274. doi: https://doi.org/10.1016/j.ins.2016.01.079

Efendi, R., Arbaiy, N., y Deris, M. M. (2018). A new procedure in stock market forecasting based on fuzzy random auto-regression time series model. *Information Sciences*, *441*, 113–132. doi: https://doi.org/10.1016/j.ins.2018.02.016

Friedman, J., Hastie, T., y Tibshirani, R. (2000). Additive logistic regression: a statistical view of boosting (With discussion and a rejoinder by the authors). *The Annals of Statistics*, *28*(2), 337–407. doi: https://doi.org/10.1214/aos/1016218223

García, F., González-Bueno, J., Guijarro, F., y Oliver, J. (2020). Forecasting the environmental,

social, and governance rating of firms by using corporate financial performance variables: A rough set approach. *Sustainability*, *12*(8), 3324. doi: https://doi.org/10.3390/su12083324

García, F., Guijarro, F., Oliver, J., y Tamošiūnienė, R. (2018). Hybrid fuzzy neural network to predict price direction in the German DAX-30 index. *Technological and Economic Development of Economy*, *24*(6), 2161–2178. doi: https://doi.org/10.3846/tede.2018.6394

Göçken, M., Özçalıcı, M., Boru, A., y Dosdoğru, A. T. (2016). Integrating metaheuristics and Artificial Neural Networks for improved stock price prediction. *Expert Systems with Applications*, *44*, 320–331. doi: https://doi.org/10.1016/j.eswa.2015.09.029

Hu, H., Tang, L., Zhang, S., y Wang, H. (2018). Predicting the direction of stock markets using optimized neural networks with Google Trends. *Neurocomputing*, *285*, 188–195. doi: https://doi.org/10.1016/j.neucom.2018.01.038

Huang, C., Yang, D., y Chuang, Y. (2008). Application of wrapper approach and composite classifier to the stock trend prediction. *Expert Systems with Applications*, *34*(4), 2870–2878. doi: https://doi.org/10.1016/j.eswa.2007.05.035

Kara, Y., Boyacioglu, M. A., y Ömer Kaan Baykan. (2011). Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the Istanbul Stock Exchange. *Expert Systems with Applications*, *38*(5), 5311–5319. doi: https://doi.org/10.1016/j.eswa.2010.10.027

Kim, K.-j. (2003). Financial time series forecasting using support vector machines. *Neurocomputing*, *55*(1-2), 307–319. doi: https://doi.org/10.1016/s0925-2312(03)00372-2

Kim, K.-j., y Han, I. (2000). Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index. *Expert Systems with Applications*, *19*(2), 125–132. doi: https://doi.org/10.1016/s0957-4174(00)00027-0

Kirkpatrick II, C. D., y Dahlquist, J. A. (2010). *Technical analysis: the complete resource for financial market technicians*. FT press.

Krauss, C., Do, X. A., y Huck, N. (2017). Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the S&P 500. *European Journal of Operational Research*, *259*(2), 689–702. doi: https://doi.org/10.1016/j.ejor.2016.10.031

Lee, M.-C. (2009). Using support vector machine with a hybrid feature selection method to the stock trend prediction. *Expert Systems with Applications*, *36*(8), 10896–10904. doi: https://doi.org/10.1016/j.eswa.2009.02.038

Leung, M. T., Daouk, H., y Chen, A.-S. (2000). Forecasting stock indices: a comparison of classification and level estimation models. *International Journal of Forecasting*, *16*(2), 173–190. doi: https://doi.org/10.1016/s0169-2070(99)00048-5

Li, D., Li, Z., y Li, R. (2017). Automate the identification of technical patterns: a k-nearest-neighbour model approach. *Applied Economics*, *50*(17), 1978–1991. doi: https://doi.org/10.1080/00036846.2017.1383596

Liaw, A., y Wiener, M. (2002). Classification and regression by randomforest. *R News*, *2*(3), 18–22.

Nguyen, T. H., Shirai, K., y Velcin, J. (2015). Sentiment analysis on social media for stock movement prediction. *Expert Systems with Applications*, *42*(24), 9603–9611. doi: https://doi.org/10.1016/j.eswa.2015.07.052

Ni, L.-P., Ni, Z.-W., y Gao, Y.-Z. (2011). Stock trend prediction based on fractal feature selection and support vector machine. *Expert Systems with Applications*, *38*(5), 5569–5576. doi: https://doi.org/10.1016/j.eswa.2010.10.079

Oppel, S., Meirinho, A., Ramírez, I., Gardner, B., O'Connell, A. F., Miller, P. I., y Louzao, M. (2012). Comparison of five modelling techniques to predict the spatial dis-

tribution and abundance of seabirds. *Biological Conservation*, *156*, 94–104. doi: https://doi.org/10.1016/j.biocon.2011.11.013

Patel, J., Shah, S., Thakkar, P., y Kotecha, K. (2015a). Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques. *Expert Systems with Applications*, *42*(1), 259–268. doi: https://doi.org/10.1016/j.eswa.2014.07.040

Patel, J., Shah, S., Thakkar, P., y Kotecha, K. (2015b). Predicting stock market index using fusion of machine learning techniques. *Expert Systems with Applications*, *42*(4), 2162–2172. doi: https://doi.org/10.1016/j.eswa.2014.10.031

Sayegh, A., Tate, J. E., y Ropkins, K. (2016). Understanding how roadside concentrations of NOx are influenced by the background levels, traffic density, and meteorological conditions using boosted regression trees. *Atmospheric Environment*, *127*, 163–175. doi: https://doi.org/10.1016/j.atmosenv.2015.12.024

Touzani, S., Granderson, J., y Fernandes, S. (2018). Gradient boosting machine for modeling the energy consumption of commercial buildings. *Energy and Buildings*, *158*, 1533–1543. doi: https://doi.org/10.1016/j.enbuild.2017.11.039

Tsinaslanidis, P. E. (2018). Subsequence dynamic time warping for charting: Bullish and bearish class predictions for NYSE stocks. *Expert Systems with Applications*, *94*, 193–204. doi: https://doi.org/10.1016/j.eswa.2017.10.055

Weng, B., Lu, L., Wang, X., Megahed, F. M., y Martinez, W. (2018). Predicting short-term stock prices using ensemble methods and online data sources. *Expert Systems with Applications*, *112*, 258–273. doi: https://doi.org/10.1016/j.eswa.2018.06.016

Zhang, J., Cui, S., Xu, Y., Li, Q., y Li, T. (2018). A novel data-driven stock price trend prediction system. *Expert Systems with Applications*, *97*, 60–69. doi: https://doi.org/10.1016/j.eswa.2017.12.026