

# Predicting stock market indices movements

Pedro N. Rodriguez<sup>1</sup> & Arnulfo Rodriguez<sup>2</sup>

<sup>1</sup>*Department of Statistics and Operations Research II, Universidad Complutense de Madrid, Spain.*

<sup>2</sup>*Department of Economics, University of Texas at Austin, USA.*

## Abstract

This paper examines the extent to which the daily movements of three large emerging markets stock indices are predictable. Lagged technical indicators are used as explanatory variables. In the analysis we employed seven classification techniques and assessed the discriminatory power of the classifiers through the area under the receiver operating characteristic (ROC) curve. The results show that the daily movements of the three indices are better predictable than random. After taking into account the bias induced by non-synchronous price quotations, a trading system with break-even costs is simulated. The non-random classifiers yield returns above those of both random walk and contrarian investment strategies. No inefficiency is found due to the fact that relatively low break-even transaction costs are enough to eliminate the sources of trading profits.

*Keywords: Stock indices movements, Data mining, ROC analysis.*

## 1 Introduction

Predicting assets returns is a subject that attracts researchers from diverse areas of investigation. Recently, a new line of research has emerged and switched the paradigm from analysing the predictability of assets returns to analysing the predictability of the direction of assets returns.

The demand for trading stock market indices has quickly grown in developing and emerging markets. Given this context, assessing the predictive power of a technique forecasting the direction of a stock index is an important objective. Success or error rates are perhaps the most popular methods to quantify the discriminatory power of a technique. It is known that the classification error criterion relies on implicit assumptions that play havoc with comparing performances both across models and within a model itself (Hand [1]).

Advances in statistics, machine learning, pattern recognition and artificial intelligence, etc. have contributed to delivering robust evaluation and classification techniques. One example is the area under the receiver operating characteristic (ROC) curve (AUC) that has been widely used to assess the discriminatory power of a classifier in medical diagnosis and imaging (Hanley and McNeil [2], Metz *et al.* [3], Zhou *et al.* [4]). The exhibition of a number of desirable properties by the AUC in contrast to the overall accuracy's explains its ground gaining (Bradley [5]).

Classification techniques that are robust to irrelevant explanatory variables and with high degrees of interpretability and predictability have been developed. The election of a technique is non-trivial due to the fact that the characteristics of the database are not known a priori. Consequently, decisions should be based on the needs of interpretation and/or prediction with computational performance taken into account. We test the extent to which the direction of several stock indices can be forecasted by using lagged technical indicators as explanatory variables (see Section 3 for details).

This article complements the existent literature in several ways. First, in addition to Neural Networks (Ripley [6]) and Logit models (Hastie *et al.* [7]), we apply recent machine learning algorithms and statistical techniques such as PolyClass (Stone *et al.* [8]), Random Forest (Breiman [9]), Gradient Boosting Machine (Friedman [10]) and Tree-based models (Breiman *et al.* [11], Breiman [12]) to the stock index movement phenomena. Second, we assess the discrimination/prediction power of the classifiers via the area under the ROC curve.

This paper is organised as follows. In Section 2, an empirical review is given. Section 3 briefly discusses the data sets, the classification techniques employed and ROC curves. Section 4 contains the out-of-sample results. Section 5 describes the trading simulation. Concluding remarks follow in section 6.

## 2 Empirical Review

The empirical literature on returns predictability testing can be categorised according to two methodological approaches. The first branch tests the predictability of assets returns. A substantial literature in this area has been accumulated up until now and its review is beyond the scope of this article. See Fama [13], Fama [14] and Campbell *et al.* [15] for complete reviews on the predictability of assets returns.

The second group of studies analyses the predictability of the direction of assets returns. The basic premise is that the direction of future returns has a direct implication on trading strategies and assets allocation.

Tsaih *et al.* [16] model the S&P 500 Stock Index Futures via a hybrid artificial intelligence system. Such hybrid system integrates a rule-based system and neural networks. Their analysis uses dichotomised technical indicators as explanatory variables. They show, at one cut-off value, that the hybrid classifier is able to discriminate an up movement from a down movement. Then they

integrate it into a trading system that outperforms the buy-and-hold investment strategy.

Zemke [17] compares the ability to forecast up-down movements at the Warsaw Stock Exchange among four Machine Learning techniques. He tests neural networks, k-nearest neighbour, naïve Bayesian classifier and Genetic Algorithm with lagged returns as explanatory variables. The k-nearest neighbour produces the best results and yields returns above index growth after being incorporated into a trading system

Leung *et al.* [18] compare the forecasting performance and investment returns of classification and level estimation models. Their result suggests that the classification models outperform the level estimation models in both tasks. By using a monthly database with macroeconomic inputs, they analyse the S&P500, FTSE 100 and Nikkei 225 stock indices. This study is the first one that attempts to use class probabilities in order to develop a multiple threshold trading strategy.

Fernandez-Rodriguez *et al.* [19] analyse the profitability of technical trading rules at the Spanish Stock Exchange. They implement a multi-layer neural network with lagged returns as inputs. The estimated neural network predicts better than a random walk directional forecast at one cut-off value.

Kim [20] employs Support Vector Machines to forecast the direction of the Korea Composite Stock Price Index. He utilizes technical indicators as explanatory variables. His results indicate that support vector machines outperform neural networks and case-based reasoning.

Chen *et al.* [21] analyse the predictability of the Taiwan Stock Exchange via probabilistic neural networks and generalised methods of moments with a Kalman filter. Their results show that the probabilistic neural networks-based investment strategies obtain higher returns than the investment strategies generated by the buy-and-hold, random walk model and parametric generalised method of moments.

These findings are of importance because they show that the relation between an accurate directional forecast of a stock index and capital gains is stronger than the one between an accurate stock index level forecast and capital gains. However, none of these studies has analysed the implications of non-synchronous closing prices on their trading strategies. Moreover, none of these studies (in the case of methods that produce continuous outputs) has analysed the prediction accuracy regardless of the cut-off value. Is a classifier better than the random model for any cut-off value? This paper addresses the aforementioned limitations.

### **3 Methodology**

#### **3.1 Data Sets**

The data sample was obtained from three large emerging markets. It consisted of daily closing index levels of the IPC (Mexico), KLSE Composite (Malaysia) and

Bovespa (Brazil) stock indices. The daily closing index levels spanned from January 1990 to December 2003 and were provided by each country's stock exchange.

This study attempts to predict the direction of the daily changes in the stock price indices. The discrete dependent variable (output) was codified with 1 when there was an up-movement, 0 otherwise.

The explanatory variables used to analyse the stock indices were the following technical indicators: the 14-day Relative Strength Index (RSI), 14-day Stochastic RSI, 4-day Momentum, 14-day Disparity, 21-day Disparity, Price Oscillator and Price rate-of-change (1&2-day). All of the explanatory variables were lagged one day. The use of these technical indicators along with their specification was based on the facts that they are widely known and commonly used with such specification. See StockCharts [27] and Chande and Kroll [22] for clear and easy explanations on the employed technical indicators.

The data were divided into two sets. One data set for training and validation and the other one for testing. About 20% of the data was used for testing. In order to avoid data snooping, we did not tweak the classification techniques parameters contingent on the error provided by the test data. All of the explanatory variables were normalised to a mean equal to zero and standard deviation equal to one. The mean and standard deviation obtained from the training data were used to normalise the test data.

### 3.2 Classification Techniques

The logistic regression model (Logit) arises from the desire to model class probabilities via linear functions in the explanatory factors space (linear decision boundaries). The Logit will produce accurate forecasts if and only if the parameterised equation is similar to the true function that discriminates up from down movements. In order to reduce the effect of irrelevant explanatory variables, a backward subset selection was carried out. See Hastie *et al.* [7] for further details.

Neural Networks model linear combinations of explanatory variables adjusted to a transfer function. The architecture of a neural network will depend on the number of explanatory variables (inputs), number of hidden layers, number of neurons per hidden layer and number of response variables (output). In this study we fixed the number of hidden layers to one and the number of neurons per hidden layer to 6 and 24. The sigmoid function was the transfer function for each layer. In the optimization process we used one fast training algorithm: the variable learning rate backpropagation with momentum. As for the stopping criteria, this study allowed 500 learning epochs since a higher number of learning epochs did not produce a decrease in the error function.

Tree-based techniques partition the explanatory variables space into a set of rectangles and then fit a simple model to each one. Tree-based model try to find the split that maximizes the decrement in the impurity function in order to make a tree grow. This is done iteratively until a certain amount of observations is reached or no further decrements in impurity functions are found. If the tree produced by the aforementioned algorithm is too large (not easy to interpret)

and/or over-fits the training data, then a pruning stage will be needed. The pruning tries to find the best ratio derived from changes in the impurity function produced by changes in the number of terminal nodes (complexity parameter). Usually, the criterion that defines the best ratio is the error rate in an independent or test sample or the k-Fold Cross-Validation error rate. See Breiman *et al.* [11] for further details.

Although tree-based models are highly popular among researchers, such models have a weakness: they produce poor estimates of class probabilities (See Provost and Domingos [23] and references therein). In order to deal with this limitation we took several strategies. First, we did not prune the classification trees. Second, we implemented a bagging predictor method. This is a method of generating multiple versions of a predictor to be used for obtaining an aggregated predictor. Each predictor is grown from datasets drawn at random (with replacement) from the training data. Each dataset is of the same size as the training set. See Breiman [12] for further details.

We used the Gini impurity function for the tree-based models. In the bagging predictor estimation, the number of bootstrap replications was incremented until the 'optimal' out-of-bag error rate (misclassification rate) was found. 'Optimal' here means that future increments in bootstrap replications did not decrease the out-of-bag error rate.

Third, we estimated a Random Forest. This technique, besides implementing a bagging predictor, uses a random selection of features to split each node (F). For each bootstrap replication, about one third of the instances are left out in the calculation of the out-of-bag error rate. See Breiman [9] for further details.

As for the Random Forest, we fixed to 1 the number of randomly selected features to split each node. The number of grown trees (bootstrap replications) was incremented until the 'optimal' out-of-bag error rate was found. 'Optimal' has the same connotation as in the bagging prediction estimation.

Finally, we estimated a Gradient Boosting Machine. This technique constructs additive regression models by fitting a simple base learner, typically a small tree, to the current 'pseudo'-residuals. The 'pseudo'-residuals are a function of the loss function employed in the analysis. In this article we used the Bernoulli loss function. Yet two questions remain: what is the right size of the tree (base learner)? And how many boosting iterations must be done? In relation to the first question, we allowed the maximum number of variables' interactions to be 1, 3 and 5. Regarding the second question, we stopped the boosting iterations when the out-of-bag estimate of the marginal reduction in the loss function was zero. With the objective to increase accuracy and execution time, we incorporated randomness in the procedure as described in Friedman [24] by using 50% of the training data to propose the next tree (base learner).

The last classification method employed in the analysis was PolyClass. It was developed by Stone *et al* [8]. The PolyClass is a hybrid of MARS (Friedman [25]) that is specifically designed to handle classification problems. Just like MARS, the PolyClass grows the model in a forward stage-wise fashion. Both methods use linear splines in their expansions. The main difference lies with the estimation algorithm. The PolyClass first looks for linear functions (without

knots), then for interactions among them and, finally, for specific knots of the function that has already been incorporated into the model. Although the decision to add each function is a function of minimizing a sum-of-squares, the final model will be estimated via the Maximum Likelihood Estimation. In order to avoid over-fitting, the model selection was carried out with the 10-Fold Cross-Validation.

### 3.3 The Receiver Operating Characteristic Curve

A ROC curve is obtained by plotting Sensitivity versus 1 - Specificity for various cut-off values. The points on a ROC curve are either joined by line segments or smooth curves with the use of nonparametric and parametric procedures, respectively.

The area under a ROC Curve (AUC) is equal to the probability that a randomly selected observation from the up-movement cases scores higher than a randomly selected observation from the down-movement cases (Hanley and McNeil [2]). A ranking probability equal to 1 and 0.5 would imply a perfect classifier and a random classifier, respectively. In this article a nonparametric approach, the Mann-Whitney-U Statistic, was chosen to obtain the AUC and its standard error was calculated with the use of the negative exponential model (Hanley and McNeil [2]).

## 4 Results

The majority of the experiments were carried out in R: Environment for Statistical Computing and Graphics with the following add-on packages: tree (developed by B.D. Ripley), ipred (developed by A. Peters and T. Hothorn), polspline (developed by C. Kooperberg), randomForest (R-port by A. Liaw and M. Wiener, original code by L. Breiman and A. Cutler) and gbm (developed by G. Ridgeway). The Neural Networks experimentation was run in MATLAB: The Language for Technical Computing, version 6.1. The AUC was estimated with ROCKit (developed by C.E. Metz).

In regard to the bagging predictor, the number of bootstrap replications for the IPC, KLSE and Bovespa stock indices was 50. The numbers of boosting iterations for the Gradient Boosting Machine with shrinkage equal to 1% for one, three and five-level interactions are shown in Table 1.

Number of boosting iterations	Index		
	IPC	KLSE	Bovespa
IL=1	13	123	72
IL=3	8	87	46
IL=5	9	84	54

Table 1: Number of boosting iterations for the Gradient Boosting Machine

In our experiments with the Random Forest, the out-of-bag error converged with less than 200 trees (bootstrap replications). The extreme case was the Bovespa index, where more than four trees substantially increased the out-of-bag error rate.

The out-of-sample discriminatory accuracy for all the classification techniques is shown in Table 2.

Techniques	Specification	AUC for each stock index		
		IPC	KLSE	Bovespa
<i>Logit</i>	Full	.5332±.0222	<b>.5726±.0222</b>	.5101±.0215
	Backwards Subset	.5260±.0222	<b>.5731±.0222</b>	.5206±.0215
<i>Neural Networks</i>	Gradient Descent with Momentum			
	6 Neurons per Hidden Layer	<b>.5545±.0221</b>	<b>.5644±.0222</b>	.5389±.0214
	24 Neurons per Hidden Layer	<b>.5512±.0221</b>	<b>.5781±.0221</b>	.5199±.0215
<i>PolyClass</i>	10-Fold Cross Validation	<b>.5498±.0221</b>	<b>.5523±.0223</b>	.5206±.0215
<i>Tree-based model</i>	Gini Impurity Function	.5239±.0222	<b>.5714±.0222</b>	.5213±.0215
	Bagging Predictor	.5430±.0222	<b>.6036±.0219</b>	.5179±.0215
<i>Gradient Boosting Machine</i>	Interaction Level (IL)			
	IL = 1	<b>.5487±.0221</b>	<b>.5600±.0223</b>	.5235±.0215
	IL = 3	<b>.5598±.0221</b>	<b>.5643±.0222</b>	.5289±.0214
	IL = 5	<b>.5654±.0220</b>	<b>.5660±.0222</b>	.5364±.0214
<i>Random Forest</i>	F=1	<b>.5503±.0221</b>	<b>.6006±.0219</b>	<b>.5537±.0213</b>

Note: Areas statistically different from one half (at 95%) in Bold.

Table 2: Area under the ROC curve and its standard error for the test data.

It can be seen from Table 2 that there was at least one classification technique per stock index that was able to discriminate better than random at the 5% significance level. Higher levels of accuracy were found in Malaysia. There all of the classification techniques were able to beat a coin toss classification. Exactly the opposite occurs in Brazil, where only one technique (Random Forest) was able to discriminate up from down movements better than random. There are two possible reasons for this discrepancy. First, the technical indicators specifications used here might not be appropriate for the Brazilian stock index. Second, the randomness injected to each split was the reason for the improved accuracy. However, the underlying favourable mechanism was not obvious.

Notice that the Random Forest was the only classification technique able to recognise a pattern in the three indices. Clearly, the results are not optimised. In the feature space one can search for the best specification of the technical indicators to improve accuracy or implement a feature extraction technique. The Tree-based models, PolyClass and Neural Networks can improve accuracy with the presence of a validation set independent of the training set. However, the results are consistent in the sense that the daily movements can be predicted better than random when using simple and common lagged technical indicators.

## 5 Trading Simulation

The evidence that some classification techniques are able to forecast the direction of the stock indices better than random suggests that it might be possible to construct a set of trading strategies. Consequently, we formulated a set of trading strategies guided by the probabilities predicted by the Gradient Boosting Machine, Random Forest and Random Forest for the IPC, KLSE composite and Bovespa stock indices, respectively.

The trading simulation assumes the following:

- Investors are able to sell short stock indices.
- Transaction costs are equal across purchases and sales of stock indices.
- Transaction costs are proportional to indices levels.

In order to avoid fixing transactional costs, we search for one-way transaction costs that make average trading profits equal to zero (Break-Even Costs).

With the objective to reduce a possible trading profits bias we followed the next strategies: First, investors can only buy or sell short stock indices. Second, trading profits or losses are made per day with no reinvestment. Third, we did not execute trades at the closing prices, instead we estimated the ‘true’ closing prices as described in Jokivuolle [26] to reduce the bias induced by non-synchronous price quotations.

Jokivuolle [26] shows that the log of the ‘true’ stock index level can be estimated by using the permanent component of the Beveridge-Nelson decomposition. We have chosen to use an MA(1) process for the IPC and KLSE for the following reasons:

- The estimated autocorrelations and partial autocorrelations were not statistically different from zero beyond the first lag.
- The Ljung-Box autocorrelation statistic for the first four lags of the MA(1) residuals was statistically insignificant.

As for the Bovespa index, we have chosen to use an MA(1) process in order to be consistent with the time series model selected for the other stock indices.

The estimated ‘true’ stock indices levels were used as the closing prices (execution prices). If the estimated probabilities are higher than the cut-off value, then he or she will buy a stock index and sell it tomorrow. If the estimated probability is lower than the cut-off value, he or she will sell short a stock index and close the position the following day. The cut-off values used in the trading simulation where the percentage of up-movements in the test sample.- i.e. in this case they were 52.5, 50 and 49 per cent for the IPC, KLSE and Bovespa indices, respectively.

The results of the trading strategies (one-way break-even transaction costs) are shown in Table 3.



<i>IPC (Mexico)</i>	B-E Costs	<i>KLSE (Malaysia)</i>	B-E Costs	<i>Bovespa (Brazil)</i>	B-E Costs
GBM(IL=5)	0.190%	Random Forest	0.039%	Random Forest	0.036%
Random Walk	-0.171%	Random Walk	0.007%	Random Walk	-0.016%
Contrarian Investment	0.171%	Contrarian Investment	-0.007%	Contrarian Investment	0.016%
Perfect Classifier	0.651%	Perfect Classifier	0.402%	Perfect Classifier	0.750%

Table 3: Break-Even (B-E) Costs.

The single day trading strategies were compared with other strategies. The random walk trading strategy will have a buy signal if the last trading day has an up-movement, and vice versa. The contrarian investment strategy will have a buy signal if the last trading day has a down-movement, and vice versa. The perfect classifier was included for comparison purposes. In other words, what are the costs that make average trading profits equal to zero despite knowing the future?

It can be seen from Table 3 that the probability-based trading models need higher one-way transaction costs than both the random walk and contrarian investment strategies. The low one-way transaction costs are mainly due to heavy trading and a relatively low discriminatory power of the classifiers.

Investment periods and different trading strategies such as switching between fixed income securities and stocks could improve profitability. However, they were not considered here because they could bias the result of a classifier only trained for understanding up-down movements and not for choosing between fixed income securities and stocks.

## 6 Conclusions

This empirical paper examined the extent to which the daily movements of three large emerging markets stock indices (Brazil, Malaysia and Mexico) were predictable. Lagged technical indicators were used as the explanatory variables of seven classification techniques. The discriminatory power of the classifiers was assessed through the area under the receiver operating characteristic (ROC) curve. The results showed that the daily movements of the three indices were better predictable than random. Indeed, the Random Forest was able to forecast the daily movements of the three stock indices better than random. After having taken into account the bias induced by non-synchronous price quotations, a trading system with break-even costs was simulated. The non-random classifiers yielded returns above those of both random walk and contrarian investment strategies. No inefficiency was found due to the fact that relatively low break-even transaction costs were enough to wipe out the sources of trading profits.

## References

- [1] Hand, D.J., *Construction and Assessment of Classification Rules*. John Wiley & Sons, Chichester. 1997.

- [2] Hanley, J.A. & McNeil, B.J., The Meaning and Use of the Area under a Receiver Operating Characteristic (ROC) Curve. *Radiology* **143** (1): 29-36, 1982.
- [3] Metz, C.E., Herman, B.A. & Shen, J., Maximum Likelihood Estimation of Receiver Operating Characteristic (ROC) Curves from Continuously-Distributed Data. *Statistics in Medicine* **17**: 1033-1053, 1998.
- [4] Zhou, X.H., McClish, D.K. & Obuchowski, N.A., *Statistical Methods in Diagnostic Medicine*, Wiley, 2002.
- [5] Bradley, A., The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition* **30** (7): 1145-1159, 1997.
- [6] Ripley, B., Neural networks and related methods for classification. *Journal of the Royal Statistical Society B* **56** (x): 409-456, 1994.
- [7] Hastie, T., Tibshirani, R. & Friedman, J.H., *The Elements of Statistical Learning: Data Mining, Inference and Prediction*. Springer-Verlag: New York, 2002.
- [8] Stone, C.J., Hansen, M., Kooperberg, C. & Truong, Y., The use of polynomial splines and their tensor products in extended linear modeling (with discussion). *Annals of Statistics* **25**: 1371-1470, 1997.
- [9] Breiman, L., Random Forest. *Machine Learning* **45** (1): 5-32, 2001.
- [10] Friedman, J.H., Greedy Function Approximation: A Gradient Boosting Machine. *Annals of Statistics* **29** (5) October, 2001.
- [11] Breiman, L., Friedman, J.H., Olshen, R.A. & Stone, C.J., *Classification and Regression Trees*. Wadsworth Inc., Pacific Grove, 1984.
- [12] Breiman, L., Bagging Predictors. *Machine Learning* **26**(2): 123-140, 1996.
- [13] Fama, E.F., Efficient Capital Markets: A review of theory and empirical work. *The Journal of Finance* **25**(2): 383-417, 1970.
- [14] Fama, E.F., Efficient Capital Markets: II. *The Journal of Finance* **46** (5): 1575-1617, 1991.
- [15] Campbell, J.Y., Lo, A.W. & MacKinlay, A.C., *The Econometrics of Financial Markets*, Princeton University Press, 1994.
- [16] Tsaih, R., Hsu, Y. & Lai, C. Forecasting S&P500 Stock index futures with a hybrid AI system. *Decision Support Systems* **23**: 161-174, 1998.
- [17] Zemke, S., Nonlinear index prediction. *Physica A* **269**: 177-183, 1999.
- [18] Leung, M.T., Daouk, H. & Chen, A.S., Forecasting stock indices: a comparison of classification and level estimation models. *International Journal of Forecasting* **16**: 173-190, 2000.
- [19] Fernandez-Rodriguez, F., Gonzalez-Martel, C. & Sosvilla-Rivero, S., On the profitability of technical trading rules based on artificial neural networks: Evidence from the Madrid stock market. *Economic Letters* **69**: 89-94, 2000.
- [20] Kim, K., Financial time series using support vector machines. *Neurocomputing* **55**: 307-319, 2003.
- [21] Chen, A.S., Leung, M.T. & Daouk, H., Applications of neural networks to an emerging financial market: forecasting and trading the Taiwan Stock Index. *Computers and Operations Research* **30**: 901-923, 2003.
- [22] Chande, T.S., & Kroll, S., *The New Technical Trader: Boost your Profits by Plugging into the Latest Indicators*, John Wiley & Sons, 1994.
- [23] Provost, F. & Domingos, P., Tree Induction for Probability-Based Ranking. *Machine Learning* **52**: 199-215, 2003.
- [24] Friedman, J.H., Stochastic Gradient Boosting, *Computational Statistics and Data Analysis* **38**(4): 367-378, 2002.
- [25] Friedman, J.H., Multivariate adaptive regression splines (with discussion), *Annals of Statistics* **19**(1) : 1-141, 1991.
- [26] Jokivuolle, E., Measuring the true stock index value in the presence of infrequent trading, *Journal of Financial and Quantitative Analysis* **30**: 455-464, 1995.
- [27] StockCharts.com, [www.stockcharts.com](http://www.stockcharts.com)