Stock Portfolio Prediction by Multi-Target Decision Support*

João Augusto Provin Ribeiro da Silva
State University of Londrina (UEL)
Computer Science Department
Londrina, Paraná, Brazil
joaoaugustoprovin@gmail.com

Everton Jose Santana
State University of Londrina (UEL)
Electrical Engineering Department
Londrina, Paraná, Brazil
evertonsantana@uel.br

Saulo Martiello Mastelini
State University of Londrina (UEL)
Computer Science Department
Londrina, Paraná, Brazil
mastelini@uel.br

Sylvio Barbon Jr.
State University of Londrina (UEL)
Computer Science Department
Londrina, Paraná, Brazil
barbon@uel.br

ABSTRACT

Investing in the stock market is a complex process due to its high volatility caused by factors as exchange rates, political events, inflation and the market history. To support investor's decisions, the prediction of future stock price and economic metrics is valuable. With the hypothesis that there is a relation among investment performance indicators, we applied multi-target regression (MTR) methods to estimate 6 different indicators aiming at creating an automated prediction tool for decision support. The experiments were based on 4 datasets, corresponding to 4 different time periods, composed of 63 combinations of weights of stock-picking concepts each, simulated in the US stock market. We compared traditional machine learning approaches with four state-of-the-art MTR solutions: Stacked Single Target, Ensemble of Regressor Chains, Deep Structure for Tracking Asynchronous Regressor Stacking and Multi-output Random Forest (MORF). With the exception of MORF, traditional approaches and the MTR methods were evaluated with Random Forest and Support Vector Machine regressors. By means of extensive experimental evaluation, our results showed that the most recent MTR solutions can achieve suitable predictive performance, improving all the scenarios (12.6% in the best period, considering all target variables). In this sense, MTR is a proper strategy for building stock market decision support system based on prediction models.

CCS CONCEPTS

• **Information systems** → Decision support systems; Retrieval tasks and goals; • **Computing methodologies** → Machine learning; Supervised learning by regression;

*Produces the permission block, and copyright information

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
SBSI'18, June 4–8, 2018, Caxias do Sul, Brazil
© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-6559-8/18/06.
https://doi.org/10.1145/3229345.3229382

KEYWORDS

Stock market; Multi-target regression; Decision support system

ACM Reference Format:
João Augusto Provin Ribeiro da Silva, Everton Jose Santana, Saulo Martiello Mastelini, and Sylvio Barbon Jr.. 2018. Stock Portfolio Prediction by Multi-Target Decision Support. In *SBSI'18: XIV Brazilian Symposium on Information Systems, June 4–8, 2018, Caxias do Sul, Brazil.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3229345.3229382

1 INTRODUCTION

The prediction of stock market is a very challenging task because it is affected by several macro-economics factors, for instance exchange rates, political events, recession or expansion periods, and investor's expectations [2, 12]. Other factors that usually influence this volatility are inflation, higher than before interest rates, rising bond yields and the stock market on it's own, since it can be overheated.[1]

When making a decision, the stock price itself is not the main information source, so that investors consider parameters that give information on return rates and the associated risks, such as price to book value ratio, dividend-price ratio, return on investment, return on equities and systematic risk [4, 10, 16, 21].

The use of decision support tools in stock trading is also helpful. These tools, as Kvout[2] and Trade Ideas[3], are becoming more sophisticated as they use artificial intelligence to improve the prediction of performance of investments.

In fact, many computational intelligence (CI) and machine learning (ML) algorithms tried to address stock forecast problem: artificial neural networks (ANNs), linear and multi-linear regression (LR, MLR), support vector machine (SVM), autoregressive moving average (ARMA) and autoregressive integrated moving average (ARIMA) models, genetic algorithms (GAs), random forest (RF) and random walk (RW) to name some [2, 3, 12, 13]. In these works, the prediction of the stock market price or economic properties were done by building ML solutions which deals with a single response or output, i.e., a single-target (ST) approach. Besides, as the stock

[1]CNNMoney (New York) (2018, February 12). How to handle stock market volatility and keep your retirement plan in check. *CNN Money.* Retrieved from http://money.cnn.com/2018/02/12/pf/applenews-stock-market-dow-down/index.html. [Accessed: 25th February 2018]
[2]www.kavout.com
[3]www.trade-ideas.com

market outcomes are continuous values, the related ML problems are called regression tasks [15].

Until now, most of the existing decision support systems modelled the stock market performance indicators as separated ST problems. Multi-target regression (MTR), however, is a research field of ML that deals with predictive problems which present multiple continuous outputs or responses, and could be better explored in stock prediction problem. In these tasks, the targets may present underlying inter-dependencies, influencing and being influenced among themselves [7, 15, 23]. Therefore, MTR aims at modelling not only the input to output relationships in a predictive problem, but also the inter-output relationships.

Our hypothesis is that the multiple stock market performance indicators may present underlying relationships among themselves. Since different performance indicators share the same explaining features, they can be modelled as a MTR problem. In this sense, a regression model for a performance indicator could use information of other outcomes to yield better predictions of itself and compose a more reliable decision support tool. Figure 1 presents an overview of two prediction methods: single-target and multi-target. This kind of prediction method composes a decision support tool in extracting knowledge from stock-picking concepts as a potential aid to the decision making of an investor.

The goal of this paper is proposing a kernel to decision support tool based on MTR to predict performance indicators of stock market, showing that this approach can generate an accurate model due to the inter-target influence in this model.

This paper is organized as follows: Section 2 presents related works, Section 3 describes the experimental setup, Section 4 reports the results and its analysis, followed by Section 5, that concludes the work.

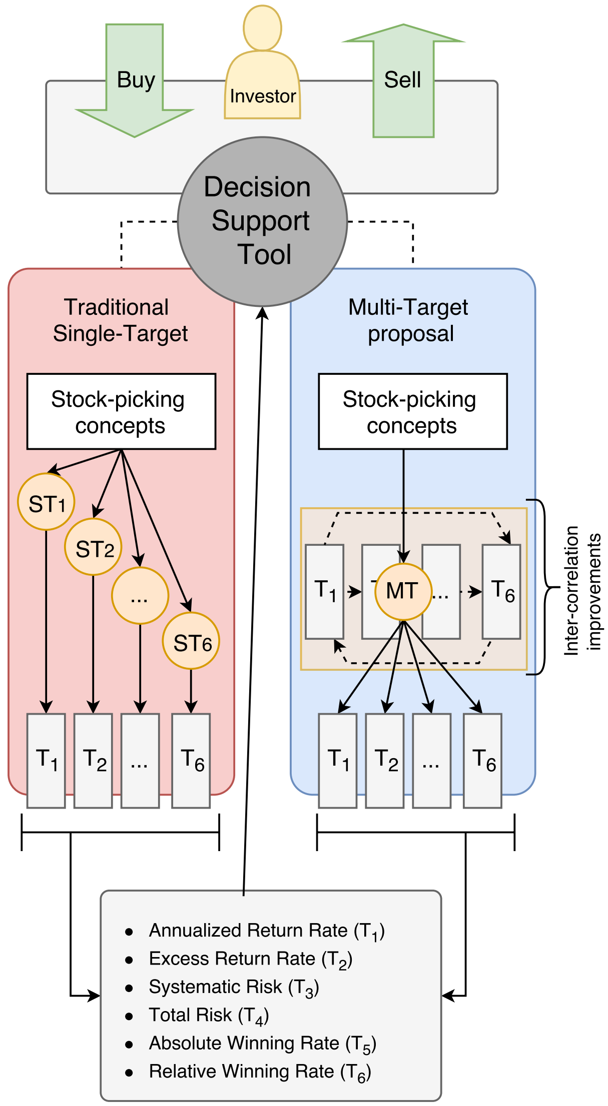

Figure 1: Decision Support Tool from stock market investor, comparing single-target and multi-target prediction kernels.

2 RELATED WORK

Machine learning algorithms have been vastly used in the literature to predict stock market characteristics. Roko and Gilli [21], for example, used classification trees improved by bootstrap aggregation to predict assets which achieve future returns above the average. According to the authors, the performance of these portfolios are significantly superior to recorded indexes.

However, the most common problems deal with continuous responses, implying in regression models. Patel et al. [20] used ANNs, SVM, RF and Naive-Bayes to predict the direction of movement of stock and stock price index for Indian stock markets and obtained as result that technical parameters as input data resulted in higher accuracy than using open, high, low and close prices.

Hu et al. [12] applied sharp ratio, a profit metric, to tune support vector regression models in stock index forecasting. The results showed that profit guided stock index forecasting is competitive and is able to produce significantly better performances than statistical error guided models.

Liu and Yeh [16] proposed the use of mixture design and ANN to build models in order to optimize weighted scoring stock selection. They found out useful models for investors to search for the optimal investment strategies to meet their specific preferences.

When it comes to MTR to stock forecasting, Xiong et al. [25] used Multiple-output support vector regression with a firefly algorithm to estimate the lower and upper values of stock price index, resulting in a promising alternative for forecasting interval-valued financial time series.

Multi-target regression intends to simultaneously predict multiple continuous variables in a common set of inputs. Furthermore, MTR provides a more appropriate interpretability of real life problems since it takes the relationship between the targets into consideration. The prediction of MT tasks had, commonly, been made through two base approaches: algorithm adaptation and problem transformation [7]. Kocev et al. [15] denotes the same strategies as global and local approaches, respectively.

The initial approach provides challenge, since it does not only aim to deal with multiple targets by changing an well known ST regression technique, but also the investigation, modelling and interpretation of the possible relations between said output variables. Through refinement methods, such as node splitting regression trees [14] and optimization functions (SVMs) [7], this approach leads to alteration in the original technique. Although complex, algorithm adaptation methods have reached satisfactory prediction performances, along with the generation of unique models and target correlation exploration [1, 7, 14, 14, 15].

Problem transformation, the latter approach, consists of data manipulation and regression techniques as means to simultaneously predict separated ST problems. Even though building separate models for each instance leads to ignoring the relationship between targets, this method may, sometimes, provide superior predictive performance and it was used as a baseline method in multiple MTR works [1, 7, 17, 23]. Nevertheless, an MTR approach carries more potential on the quality of predictions due to target dependence exploration.

Throughout the years, some problem transformation methods were created, aiming at exploring inter-target dependencies through the employment of multiple ST regressors [7, 17–19, 22–24]. Among them, some multi-label classification methods began seeing adaptations to MTR. As proposed in Spyromitros-Xioufis et al. [23], two relevant methods, inspired by the previously mentioned research area, came to fruition: SST (Stacked Single-Target) and ERC (Ensemble of Regressor Chains). They also influenced some posterior researches in MTR [17–19, 22]. To the best of our knowledge, problem transformation multi-target methods were never used to predict performance indicators of a stock portfolio.

The SST method consists of two major steps. It begins by training separate d ST models, where d represents the number of targets. However, the important part lies within the second stage, where instead of directly using these models for prediction, the method provides an additional training step using each target, thus generating d meta-models.

In essence, SST implements the idea of correcting the predictions acquired throughout the first stage, thus increasing the task's description capability and the prediction performance with the insertion and exploration of target correlation.

The ERC method makes use of target chains, which are randomly chosen to form a set. By following the chain sequence, ST regression models are formed for each target, and then trained following the order of the sequence. The method creates new datasets with the combination of the native variables and the predictions of the last models. After repeating this process for the whole chain sequence, the training is done. New instances shall be directed to the set of chains. Then, generating the average result of the predicted y values should return the final prediction for the y target. Permutation is an important part of the whole process. ERC makes use of all target combinations, as long as the number of total permutations is equal to or less than 10. If that is not the case, then 10 combinations are selected.

Deep Structure for Tracking Asynchronous Regressor Stacking (DSTARS) was recently proposed by Mastelini et al. [17] as an extension to the original SST idea. The authors proposed the employment of multiple steps of regressor stacking for each target in a dynamic way. Their hypothesis is that the addition of more stacked regressors could decrease the prediction error of the most dependent targets. In this sense, differently from the previous methods, DSTARS considers multiple levels of inter-target dependencies explicitly. The authors reported superior results than the SST and ERC methods in MTR benchmarking problems.

The Multi-output Random Forest (MORF) method is an extension to the original RF formulation and represents an algorithm adaptation (global) method. Instead of using regular regression tree algorithms [8, 9], MORF employs Predictive Clustering Trees (PCT), which were proposed by Blockeel et al. [6]. PCTs create at each data partition clusters containing all instances which match the evaluated decisions made. The root node corresponds to the cluster containing all the training instances. PCT can be seen as a generalization of traditional decision tree algorithms. The PCTs which compose MORF are grown by creating clusters which aim at minimizing the intra-cluster variances, while increasing the inter-cluster variance. The predictions made for each target correspond to the average of all instances which lie in the considered leaf node. MORF was employed in multiple MTR works [1, 7, 23] as a comparison method in past years, and it was also use in our experiments.

3 EXPERIMENTAL SETUP

This section presents the datasets used in this work, as well as the chosen regression techniques and evaluation metrics.

3.1 Datasets

The datasets used in this work were based on simulations with Standard and Poor's Compustat US database in four periods from 1990 to 2010 [16]: the first from September of 1990 to June of 1995, the second from September of 1995 to June of 2000, the third from September of 2000 to June of 2005 and the fourth from September of 2005 to June of 2010.

Each of these periods consists of 63 different weighting combinations, selected from the top 10% overall weighted scores.

It has as feature inputs six different weights of the stock-picking concepts:

(1) the large book value-to-price ratio (B/P),
(2) large sales-to-price ratio (S/P);
(3) large return on equity (ROE);
(4) large return rate in the last quarter, used to select stocks with high return rates;
(5) large market capitalization, used to select stocks with high liquidity and low risk;
(6) small systematic risk, used to select stocks with low risk in the next holding period.

The outputs are 6 investment performance indicators:

(1) Annualized Return Rate (ARR), calculated by $((1 + R)^{1/t}) - 1$, with t corresponding to period (in years) and R to the accumulated return rates,
(2) Excess Return Rate (α),
(3) Systematic Risk (β), that are coefficients of the regression $R_i - R_f = \alpha + \beta(R_m - R_f)$, where R_i is the investment portfolio return rate, R_f the risk-free return rate and R_m the market return rate,
(4) Total Risk (TR), corresponding to the standard deviation of the return rate of the portfolio during a certain period,

(5) Absolute Winning Rate (AWR), that is the ratio between the number of portfolio holding periods with positive return rate and the total number of portfolio holding periods.
(6) Relative Winning Rate (RWR), that is the ratio between the number of portfolio holding periods with a return rate greater than the market return rate and the total number of portfolio holding periods.

Figure 2 presents the linear correlation coefficients (Pearson coefficient) calculated over all the targets. The closer to one (or negative one) the coefficient, the more correlated (or inversely correlated) are the compared targets. We have comprised all the considered periods within a single set to measure how the targets relate to each other. Although the observed correlations correspond only to linear relationships, additional nonlinear inter-dependencies may be explored by MTR methods.

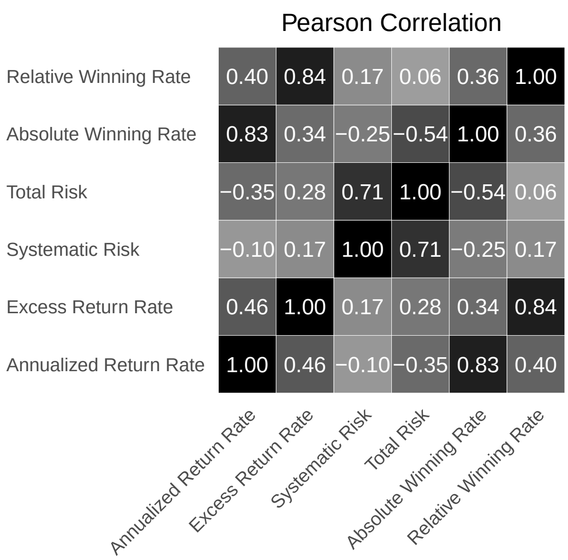

Figure 2: Pearson coefficient calculation for all the evaluated targets

The existence of dependencies among the targets should lead the MTR methods to increased their predictive performance when compared to the ST strategy. As presented in Figure 2, prominent relationships were observed for some targets. For instance, the Annualized Return Rate is highly correlated with the Absolute Winning Rate, as well as the Excess Return Rate is related to the Relative Winning Rate. Some inverse relationships were also observed, such as, the Total Risk and the Absolute Winning Rate.

3.2 Regression Techniques

The experiments made use of two machine learning regression algorithms: Random Forest (RF) and Support Vector Machine (SVM). We aimed at evaluating techniques belonging to different ML paradigms. All of the algorithms had their implementations in the R programming language and used standard parameter settings.

Random Forest: As the name suggests, the RF [8] algorithm makes use of multiple decision trees [9]. When applied to regression, the tree predictors do not take on class labels, but actually continuous numerical values. The bagging meta-algorithm [8] is used to grow trees, making different training datasets be formed by bootstrap sampling. By taking the average results of all trees in the Forest, the RF predictor is formed. In our experiments we used the `ranger` R package.

Support Vector Machine: Through the use of support vectors, SVM is a kernel-based technique for classification and regression. Its main idea is to fit a decision hyperplane which adapts to the dealt problem characteristics [5]. The modeling of nonlinear tasks is done by transforming the input space. To do so, a kernel function is applied to the input variables, possibly increasing their dimension. By changing the characteristics of the input data, data separability also increases. This technique was executed in R through the `e1071` package.

3.3 Evaluation metrics

The evaluation was conducted performing a 10-folds cross-validation strategy. We chose three metrics to evaluate the performance of the compared MTR methods and regression techniques on the presented datasets. The mentioned metrics were the Relative Root Mean Square Error (RMMSE), the average Relative Root Mean Square Error (aRRMSE) and the Relative Performance (RP) [7, 23]. Besides, we also reported the percentage of error reduction of the local MTR methods over the ST strategy, considering all targets.

The RRMSE compares the predictive error obtained by a regressor when compared with the performance of a simple predictor which always outputs the target mean. In this sense, this metric measures how a predictor was capable of learning the distribution of the evaluated data by comparing it to a baseline regressor. Every time the error obtained by a regressor is very close to the referred mean predictor, the resulting RRMSE will tend to one. If the evaluated regressor performs worse than the mean predictor, the resulting RRMSE will be greater than one. The RRMSE calculation is given by:

$$\mathrm{RRMSE}(y,\hat{y}) = \sqrt{\frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \overline{y})^2}},$$

where y_i and $\hat{y}_i$ represent, respectively, the true and the predicted values for the ith instance of the target y. Besides, $\overline{y}$ represents mean value of the target y, and N represents the number of evaluated problem's instances.

The aRRMSE is calculated by averaging the RRMSE of all targets Y. The aRRMSE calculation is given by:

$$\mathrm{aRRMSE}(Y,\hat{Y}) = \frac{1}{d}\sum_{y,\hat{y}}^{Y,\hat{Y}} \mathrm{RRMSE}(y,\hat{y}),$$

where $\hat{Y}$ represents the predictions obtained for Y, and d is the number of targets.

Next, RP is obtained by dividing the aRRMSE of the ST approach by the corresponding error metric of another MTR method, as chosen in the next expression:

$$\mathrm{RP_M} = \frac{\mathrm{aRRMSE_{ST}}}{\mathrm{aRRMSE_M}},$$

where M represents an MTR method. In our case, we compared the performance of SST, ERC and DSTARS with the ST approach to verify whether the use of a MTR method could reduce the error when forecasting stock indexes. A RP greater than one implies that the chosen MTR method surpassed the ST approach, whereas outcomes smaller than one indicate that ST approach was the better choice.

Last, we employed the Friedman test to verify whether an MTR method was statistically better than the others, using a confidence level, α, of 0.05. Anytime the obtained p-values were smaller than α, we performed the post hoc Nemenyi test to rank the compared MTR methods. We graphically represented the obtained ranks, as proposed by Demšar [11]. In this representation, MTR approaches connected by a Critical Distance (CD) value are statistically equivalent, with respect to the chosen confidence level.

4 RESULT AND DISCUSSION

We performed all the mentioned MTR methods over the four stock index datasets using the chosen regression techniques. Our goal was to evaluate whether the MTR solutions would achieve satisfactory predictions results, enabling the creation of a decision support system to help in predicting the tendencies in the action market.

The obtained aRRMSE values are reported in Table 1. In this table, the smallest errors obtained per regression technique are in bold, whereas the smallest ones per dataset are underlined. MORF presents just a single result per dataset since it is a global MTR method.

Table 1: The aRRMSE results obtained from all the compared MTR approaches and regression techniques

Dataset	Regressor	ST*	SST	ERC	DSTARS	MORF
1st period	RF	0.7511	**0.7129**	0.7467	0.7137	0.9291
	SVM	0.6827	<u>**0.6316**</u>	0.6481	0.6331	
2nd period	RF	0.8017	**0.7553**	0.7802	0.7706	0.9656
	SVM	0.7417	0.7269	<u>**0.7145**</u>	0.7260	
3rd period	RF	0.7270	0.6716	0.6886	**0.6616**	0.9171
	SVM	0.5860	0.5224	0.5492	<u>**0.5204**</u>	
4th period	RF	0.8307	0.7667	0.7848	**0.7549**	1.0887
	SVM	0.7384	<u>**0.7064**</u>	0.7138	0.7098	

* ST refers the traditional machine learning

As it can be observed, SVM obtained the best result overall when compared with the RF regressor, which is also the base of the MORF algorithm. Despite no fine tuning was performed, SVM achieved the smallest error rates for all evaluated datasets. When developing a final product with the best set of MTR tools, further adjustments could be made to improve even more the performance of the mentioned regressor.

Regarding the MTR methods, SST presented the best results in four out eight times; DSTARS obtained the smallest errors three times and ERC was the best method once. Considering that, SST, which is the most simple approach apart from the ST, would be a reliable and lightweight choice to compose a decision making system. Nevertheless, the smallest error among all the datasets was obtained by DSTARS in the third observed period. This dataset consists of observations made in the period between 2000 and 2005, and presented prominently smaller aRRMSE than the other periods.

Moreover, it is worth mentioning that DSTARS achieved results very close to the SST ones in almost all cases, which is explained by the nature of this method. In fact, DSTARS is an extension of the original SST idea, being built upon stacking multiple regressors for each target. Potentially DSTARS can mimic the SST behaviour, which explains the very similar results obtained by the two methods. Notwithstanding, SST performed slightly better and it is more lightweight than DSTARS, so this method ought to present reliable and fast responses in real scenarios. Besides, the global MTR method MORF presented the worst results in all cases.

Next, we compared the performance of the local MTR methods against the ST approach. Our objective was to verify whether the exploration of the possible existing inter-dependencies among the stock index outputs led to prediction performance improvements. In this sense, we employed the RP to compared SST, ERC and DSTARS to the ST approach. The obtained results are summarized in the Table 2.

Table 2: RP values for SST along the four periods

Method	Regressor	1st	2nd	3rd	4th
SST	RF	1.05349	1.06141	1.08235	1.08358
	SVM	1.08100	1.02030	1.12178	1.04533
DSTARS	RF	1.05235	1.04035	1.09874	1.10039
	SVM	1.07849	1.02150	1.12600	1.04033
ERC	RF	1.00578	1.02756	1.05570	1.05858
	SVM	1.05339	1.03800	1.06710	1.03452

As shown in the table, all compared MTR methods obtained RPs greater than one in all cases, meaning that they outperformed the ST approach. Moreover, DSTARS presented the highest average RP (1.06977), followed by SST (1.06866). ERC presented a strictly smaller average RP than the other approaches (1.04258), reflecting the previously presented results (refer to Table 1).

Table 3 presents the percentage of error reduction obtained by the MTR when compared to the ST strategy. Due to space limitation, we reported only the results obtained by SVM, since it presented the smallest errors overall. As it can be seen, there were gains in almost all cases. The greatest gain in prediction performance was obtained by SST when predicting the ARR in the 3rd period (23.06% of error reduction). Notwithstanding, there were some cases where the MTR methods were surpassed by the ST strategy. At the worst case, the MTR methods were 3.71% worse than ST (SST for target RWR, 2nd period). In general, the smallest gains were achieved for targets TR and AWR, regardless the considered period. In general, when the MTR methods were inferior to the ST strategy regarding predictive performance, DSTARS presented the smallest degradation. In one case, it tied with ST (target RWR, 2nd period), whereas SST and ERC were 3.71% and 1.13% worse than the simplest method, respectively. Again, these observations are related with the nature of DSTARS, as previously discussed.

We also performed statistical tests to verify the possible significant statistical superiority of some MTR method in the evaluated

Table 3: Percentage of error reduction achieved by the MTR methods over the ST strategy, when using SVM as regressor. The best gains in prediction performance per dataset are in bold, whereas the worst ones are in italic

Period	Target	SST	ERC	DSTARST
1st	ARR	**14.32%**	11.73%	10.47%
	α	13.91%	9.04%	9.97%
	β	13.13%	6.42%	10.45%
	TR	5.64%	5.81%	6.18%
	AWR	4.81%	*3.11%*	8.24%
	RWR	7.16%	3.66%	6.60%
2nd	ARR	7.04%	6.98%	4.66%
	α	0.52%	4.62%	0.00%
	β	**11.40%**	7.17%	7.25%
	TR	5.67%	3.95%	4.83%
	AWR	1.56%	2.90%	-0.36%
	RWR	*-3.71%*	-1.13%	0.00%
3rd	ARR	**23.06%**	5.87%	22.02%
	α	22.02%	15.37%	21.24%
	β	5.41%	*2.94%*	6.73%
	TR	4.18%	4.55%	5.98%
	AWR	6.54%	3.77%	10.52%
	RWR	13.51%	7.64%	13.37%
4th	ARR	1.88%	3.18%	-2.62%
	α	4.19%	3.86%	2.83%
	β	**10.90%**	8.59%	10.71%
	TR	8.74%	5.29%	10.88%
	AWR	-0.97%	*-1.13%*	-0.62%
	RWR	9.83%	7.92%	8.32%

problems. We considered multiple comparison scenarios, using both the aRRMSE and RRMSE metrics, to evaluate the performance of the MTR methods in a general way and specifically to each target, respectively. Besides, for some evaluations, we focused our comparisons on the SVM regressor, since this technique obtained the smallest aRRMSE in all datasets.

Firstly, we considered the aRRMSE results obtained by all MTR approaches and all evaluated regressors. Figure 3 presents the Nemenyi results observed in this scenario. As expected, the first positions were filled by the local methods when coupled with the SVM regressor. SST appeared in the first position, being followed by DSTARS, ERC and ST, in this order. There was no statistically significant differences among the elements of this first group, but ST performed visibly worse than the other methods. The same ranking positions were observed when considering the RF regressor, which was inferior to SVM, as previously stated. However, this time, the differences between SST and DSTARS were minimal, which derivates from the nature of these methods, as we discussed previously. MORF was significantly worse than the other approaches.

Next, we compared only the local MTR approaches when combined with SVM. So, we considered only the aRRMSE of the elements observed in the first ranking group of the previous comparison. Figure 4 presents the obtained test results. Again, SST appeared in the first position, being statistically equivalent to DSTARS. The latter, in its turn, was equivalent to ERC. The ST approach was

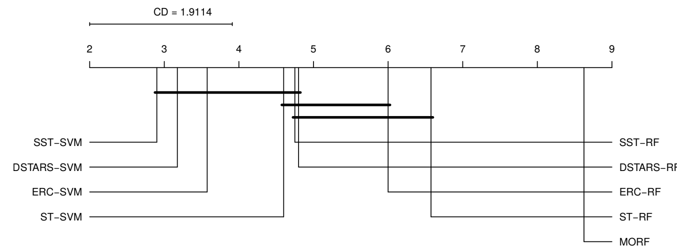

Figure 3: Nemenyi post hoc test aRRMSE results when considering all the MTR approaches and regression techniques

significantly worse than the MTR methods, which reinforces the evidences of statistical dependencies among the stock index output variables. In fact, the obtained results show that evaluated problems are MTR tasks, and thus, they must be modelled in this way to achieve superior performance.

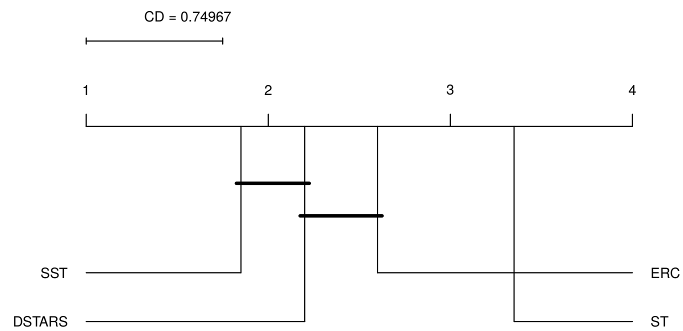

Figure 4: Nemenyi post hoc test aRRMSE results when considering all the local MTR approaches along with SVM

Following, we evaluated the obtained RRMSE results to compare the MTR approaches performance regarding each target. Figure 5 presents the Nemenyi test results when considering the RRMSE of all the evaluated MTR approaches and regression techniques. Again, the best results were obtained by the SVM regressor. Regarding the MTR methods, SST was the best approach, followed by DSTARS and ERC. The combination ST-SVM almost tied with the variations of SST and DSTARS when they were coupled with the RF regressor. Also, these three mentioned approaches were equivalent to ERC-SVM. In fact, in this comparison, ERC was not statistically better than the ST approach, regardless the chosen regressor. MORF was the worst method, being statistically worse than the other methods, as expected.

Lastly, we evaluated how the local MTR methods performed regarding the RRMSE. Again, MORF was not added in this comparison due to its marginal results. Moreover, only the SVM regressor was considered, since this technique obtained the smallest error values in all cases. Figure 6 summarizes the obtained results. The ST approach achieved the last position in the ranking, being statistically inferior to the other methods. This result reinforces the evidence that the dealt problems present inter-target dependencies, which were explored by the MTR methods. In this comparison, SST was equivalent to DSTARS, which in its turn, did not differ from ERC.

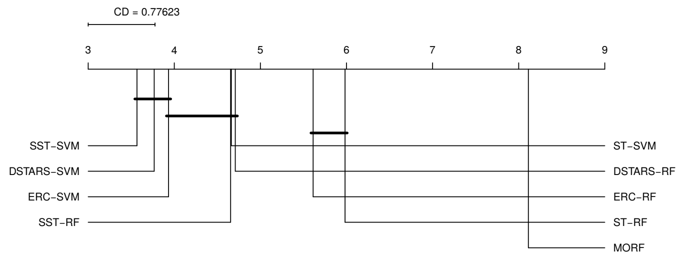

Figure 5: Nemenyi post hoc test RRMSE results when considering all the MTR approaches and regression techniques

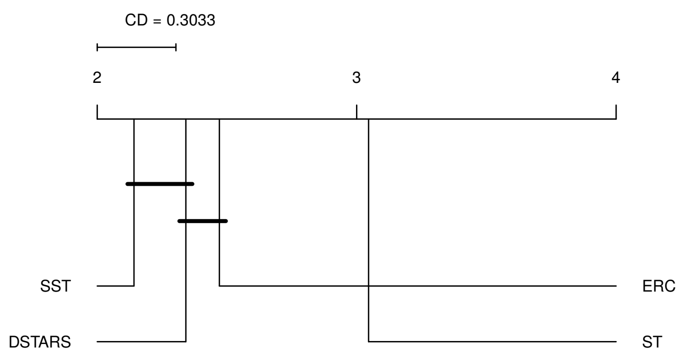

Figure 6: Nemenyi post hoc test RRMSE results when considering all the local MTR approaches along with SVM

As showed in the previously presented analysis, the employment of MTR techniques to model the prediction of multiple stock market variables, indeed, obtained superior results than considering each output separately. Among the evaluated MTR methods, SST presented the best results in most of the cases, besides being a simple yet effective solution. In our experiments, SVM outperformed RF when both techniques were implemented without further adjustments. In this sense, to compose a final solution, the combination of SST and ST would be a good choice when considering the prediction performance and the solution's computational cost.

With enough training instances, a software to offer stock indexes estimates can be created. Besides, as the time evolves and the real outcomes come to knowledge, the prediction models can be updated. A further venue for research would be the comparison of two model update strategies: the selection of training instances with a sliding window of time or aggregating all new cases through in a single knowledge database.

5 CONCLUSION

In this paper, we proposed the use of Multi-Target approaches for improving the prediction of the stock market. Moreover, it plays like a kernel of a decision support tool to aid the investor's decisions with superior results when compared to traditional solutions based on Machine Learning.

Our contribution to stock target prediction overperformed the traditional single target methods in a real-life scenario using two different learn-based algorithms. In all cases, the MT solution was superior to ST, reaching 12.6% of improvements in the third period, the best one. In this way, we affirm that the use of Multi-target improves the capabilities of stock market predictions taking advantage from their inter-correlations to built a predictor.

For future work, we aim at dealing with online updating of model owing to check concept-drifts on a data stream scenario, focusing on standard behaviour changing over the time.

ACKNOWLEDGMENTS

We would like to thank the Brazilian agencies CAPES (*Coordenação de Aperfeiçoamento de Pessoal de Nível Superior*) and CNPQ (*Conselho Nacional de Desenvolvimento Científico e Tecnológico*) for scholarships.

REFERENCES

[1] T Aho, B Zenko, S Dzeroski, and T Elomaa. 2012. Multi-Target Regression with Rule Ensembles. *J. Mach. Learn. Res.* 13 (2012), 2367–2407.

[2] George S. Atsalakis and Kimon P. Valavanis. 2009. Surveying stock market forecasting techniques – Part II: Soft computing methods. *Expert Systems with Applications* 36, 3, Part 2 (2009), 5932 – 5941.

[3] Arash Bahrammirzaee. 2010. A comparative survey of artificial intelligence applications in finance: artificial neural networks, expert system and hybrid intelligent systems. *Neural Computing and Applications* 19, 8 (01 Nov 2010), 1165–1195.

[4] S. Basu. 1977. INVESTMENT PERFORMANCE OF COMMON STOCKS IN RELATION TO THEIR PRICE-EARNINGS RATIOS: A TEST OF THE EFFICIENT MARKET HYPOTHESIS. *The Journal of Finance* 32, 3 (1977), 663–682.

[5] Asa Ben-Hur and Jason Weston. 2010. A user's guide to support vector machines. *Data Mining Techniques for the Life Sciences* (2010), 223–239.

[6] Hendrik Blockeel, Luc De Raedt, and Jan Ramon. 1998. Top-Down Induction of Clustering Trees. In *Proceedings of the Fifteenth International Conference on Machine Learning*. Morgan Kaufmann Publishers Inc., 55–63.

[7] Hanen Borchani, Gherardo Varando, Concha Bielza, and Pedro Larrañaga. 2015. A survey on multi-output regression. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 5, 5 (2015), 216–233.

[8] Leo Breiman. 2001. Random forests. *Machine learning* 45, 1 (2001), 5–32.

[9] Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. 1984. *Classification and regression trees.* CRC press.

[10] Renato Bruni. 2017. Stock Market Index Data and indicators for Day Trading as a Binary Classification problem. *Data in Brief* 10 (2017), 569 – 575.

[11] Janez Demšar. 2006. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research* 7 (2006), 1–30.

[12] Zhongyi Hu, Yukun Bao, Raymond Chiong, and Tao Xiong. 2017. Profit guided or statistical error guided? A study of stock index forecasting using support vector regression. *Journal of Systems Science and Complexity* 30, 6 (2017), 1425–1442.

[13] Luckyson Khaidem, Snehanshu Saha, and Sudeepa Roy Dey. 2016. Predicting the direction of stock market prices using random forest. *CoRR* abs/1605.00003 (2016).

[14] Dragi Kocev, Celine Vens, Jan Struyf, and Sašo Džeroski. 2007. Ensembles of multi-objective decision trees. In *European Conference on Machine Learning*. Springer, 624–631.

[15] Dragi Kocev, Celine Vens, Jan Struyf, and Sašo Džeroski. 2013. Tree ensembles for predicting structured outputs. *Pattern Recognition* 46, 3 (2013), 817–833.

[16] Yi-Cheng Liu and I-Cheng Yeh. 2017. Using mixture design and neural networks to build stock selection decision support systems. *Neural Computing and Applications* 28, 3 (01 Mar 2017), 521–535.

[17] Saulo Martiello Mastelini, Everton Jose Santana, Ricardo Cerri, and Sylvio Barbon. 2017. DSTARS: A Multi-target Deep Structure for Tracking Asynchronous Regressor Stack. In *2017 Brazilian Conference on Intelligent Systems (BRACIS)*. IEEE, 19–24.

[18] Gabriella Melki, Alberto Cano, Vojislav Kecman, and Sebastián Ventura. 2017. Multi-target support vector regression via correlation regressor chains. *Information Sciences* 415 (2017), 53–69.

[19] Jose M Moyano, Eva L Gibaja, and Sebastián Ventura. 2017. An evolutionary algorithm for optimizing the target ordering in Ensemble of Regressor Chains. In *Evolutionary Computation (CEC), 2017 IEEE Congress on*. IEEE, 2015–2021.

[20] Jigar Patel, Sahil Shah, Priyank Thakkar, and K Kotecha. 2015. Predicting stock and stock price index movement using Trend Deterministic Data Preparation and machine learning techniques. *Expert Systems with Applications* 42, 1 (2015), 259 – 268.

[21] I. Roko and M. Gilli. 2008. Using economic and financial information for stock selection. *Computational Management Science* 5, 4 (01 Oct 2008), 317–335.

[22] Everton J. Santana, Saulo M. Mastelini, and Sylvio Barbon Jr. 2017. Deep Regressor Stacking for Air Ticket Prices Prediction. In *XIII Brazilian Symposium on Information Systems: Information Systems for Participatory Digital Governance*. Brazilian Computer Society (SBC), 25–31.

[23] Eleftherios Spyromitros-Xioufis, Grigorios Tsoumakas, William Groves, and Ioannis Vlahavas. 2016. Multi-target regression via input space expansion: treating targets as inputs. *Machine Learning* 104, 1 (2016), 55–98.

[24] Grigorios Tsoumakas, Eleftherios Spyromitros-Xioufis, Aikaterini Vrekou, and Ioannis Vlahavas. 2014. Multi-target regression via random linear target combinations. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 225–240.

[25] Tao Xiong, Yukun Bao, and Zhongyi Hu. 2014. Multiple-output support vector regression with a firefly algorithm for interval-valued stock price index forecasting. *Knowledge-Based Systems* 55 (2014), 87–100.