Received: 7 November 2015 | Revised: 7 April 2016 | Accepted: 8 July 2016

DOI 10.1111/exsy.12169

WILEY Expert Systems

ARTICLE

# A comparative study of approaches to forecast the correct trading actions

Luís Baía[1,2] | Luís Torgo[1,2]

[1] LIAAD - INESC TEC, Porto, Portugal

[2] Departamento de Ciência de Computadores-Faculdade de Ciências, Universidade do Porto, Porto, Portugal

**Correspondence**
Luís Baía, LIAAD-INESC TEC, Porto, Portugal.
Email: luisbaia_1992@hotmail.com

## Abstract

This paper addresses the problem of decision making in the context of financial markets, more specifically, the problem of forecasting the correct trading action for a certain future horizon. We study and compare two alternative ways of addressing these forecasting tasks: (a) using standard numeric prediction models to forecast the variation on the prices of the target asset and, on a second stage, transform these numeric predictions into a decision according to some predefined decision rules; and (b) use models that directly forecast the right decision thus ignoring the intermediate numeric forecasting task. The objective of our study is to determine if both strategies provide identical results or if there is any particular advantage worth being considered that may distinguish each alternative in the context of financial markets.

**KEYWORDS**

classification, extensive experimental comparison, forecast, regression, trading actions

## 1 | INTRODUCTION

Many real-world applications require decisions to be made based on forecasting some numeric quantity. Sales forecasting may lead to some important decisions concerning the production process. Asset price forecasting may lead investors to buy or sell some financial product. Forecasting the future evolution of some indicator of a patient may lead a medical doctor to prescribe some important treatments. These are just a few examples of concrete applications that fit this general setting: decisions based on numeric forecasts of some variable. Frequently, the decision process is based on a predefined protocol that associates intervals of the range of the numeric variable with concrete actions/decisions. This means that once we have a prediction for the numeric variable, we use some deterministic process to reach the action/decision to be made. In spite of the generality of this type of applications, this work is focused on analyzing them in the context of financial markets. In this domain the goal of investors is to make the correct trading decision (Sell, Buy, or Hold) at any given point in time. These decisions are made based on the investor's expectations on the future evolution of the asset prices. In this work, we approach this decision problem using prediction models. More specifically, we will compare two possible ways of trying to forecast what is the correct trading decision at any point in time.

In our target applications, we assume that there are deterministic decision rules that given the estimated evolution of the prices of the asset will indicate the trading action to be taken. These rules are typically driven by the investor's preferences concerning important aspects like financial risk. For instance, a rule could state that if the forecast of the variation of prices is a 2.5% increase then the correct decision is to buy the asset as this will allow covering transaction costs and still have some profit. Given the deterministic mapping from forecasted values into decisions, we can define the prediction task in two ways. The first consists on obtaining a numeric prediction model that we can use to obtain predictions of the future variation of the prices, which are then transformed (deterministically) into trading decisions (e.g., Hellstrom (1999); Lu, Lee, and Chiu (2009)). The second alternative consists of directly forecasting the correct trading decisions (e.g., Luo and Chen (2013); Ma, Song, Hung, Su, and Huang (2012); Teixeira and de Oliveira (2010)). Which is the best option in terms of financial results? To the best of our knowledge, no comparative study was carried out to answer this question. This is the goal and the main contribution of this paper: to compare these two approaches to decision making and provide experimental evidence of the advantages and disadvantages of each alternative.

## 2 | PROBLEM FORMALIZATION

The problem of decision making based on forecasts of a numerical (continuous) value can be formalized as follows. We assume there is

an unknown function that maps the values of $p$ predictor variables into the values of a certain numeric variable $Y$. Let $f$ be this unknown function that receives as input a vector $\boldsymbol{x}$ with the values of the $p$ predictors and returns the value of the target numeric variable $Y$ whose values are supposed to depend on these predictors,

$$f : \mathbb{R}^p \rightarrow \mathbb{R}$$
$$\mathrm{x} \mapsto f(\mathrm{x}).$$

We also assume that based on the values of this variable $Y$ some decisions need to be made. Let $g$ be another function that given the values of this target numeric variable transforms them into actions/decisions,

$$g : \mathbb{R} \rightarrow \mathcal{A} = \{\alpha_1, \alpha_2, \alpha_3 \ldots\}$$
$$Y \mapsto g(Y).$$

where $\mathcal{A}$ represents a set of possible actions.

In our target applications, functions $f$ and $g$ are very different. Function $g$ is known and deterministic, in the sense that it is part of the domain background knowledge. Function $f$ is unknown and uncertain. The only information we have about function $f$ is a historical record of mappings from $\boldsymbol{x}$ into $Y$, that is, a data set that can be used to learn an approximation of the function $f$. Given that the variable $Y$ is numeric, this approximation could be obtained using some existing multiple regression tools. This means that given a data set $D_r = \{\langle \mathrm{x}_i, Y_i \rangle_{i=1}^n\}$, we can use some regression tool to obtain a model $\hat{r}(\mathrm{x})$ that is an approximation of $f$. From an operational perspective, this would mean that given a test case $\boldsymbol{q}$ for which a decision needs to be made, we would proceed by first using $\hat{r}$ to obtain a prediction for $Y$ and then apply $g$ to this predicted value to get the predicted action/decision, that is, $\boldsymbol{q} \mapsto \hat{r}(\boldsymbol{q}) \mapsto g(\hat{r}(\boldsymbol{q}))$. In the context of financial markets, the predictors describe the currently observed dynamics of the prices of some financial asset, and the target numeric variable $Y$ represents the future variation of this price. This means that $f$ is the unknown function that maps the currently observed price dynamics into a future evolution of the price. On the other hand, $g$ is a deterministic function (typically based on domain knowledge and risk preferences of traders) that maps the prediction of the future evolution of prices into one of three possible decisions: Sell, Hold, or Buy.

Given the deterministic nature of $g$, we can use an alternative process for obtaining decisions. More specifically, we can build an alternative data set $D_c = \{\langle \boldsymbol{x}_i, g(Y) \rangle_{i=1}^n\}$, where the target variable is the decision associated with each known $Y$ value in the historical record of data. This means that we have a nominal target variable, that is, we are facing a classification task. Once again, we can use some standard classification tool to obtain an approximation $\hat{c}$ of the unknown function that maps the predictors into the correct actions/decisions. Once such model is obtained, we can use it given a query case $\boldsymbol{q}$ to directly estimate the correct decision by applying the learned model to the case, that is, $\boldsymbol{q} \mapsto \hat{c}(\boldsymbol{q})$. This means that given the description of the current dynamics of the price, we will use function $\hat{c}$ to forecast directly the correct trading action for this context.

Independently, of the approach followed, the final goal of the applications we are targeting is always to make correct decisions. This means that whatever process we use to reach a decision, it will be evaluated in terms of the "quality" of the decisions it generates. In this context, it seems that the classification approach, by having as target variable the decisions, would be easier to bias towards optimal actions. However, this approach completely ignores the intermediate numeric variable that is supposed to influence decisions, though one may argue that information on the relationship between $Y$ and the decisions is "encoded" when building the training set $D_c$ by using as target the values of $g(Y_i)$. On the other hand, although the regression approach is focused on obtaining accurate predictions of $Y$, it completely ignores questions like eventual different cost/benefits of the different possible decisions that could be easily encoded into the classification tasks. All these potential trade-offs motivate the current study. The main goal of this paper is to compare these two approaches in the context of financial markets.

## 3 | MATERIAL AND METHODS

This section describes the main issues involved in the experimental comparison we will carry out with the goal of comparing the two possible approaches described in the previous section.

### 3.1 | The tasks

The problem addressed in this paper is very common in automatic trading systems where decisions are based on the forecasts of some prediction models. The decisions to open or close short/long positions are typically the result of a deterministic mapping from the predicted prices variation.

In our experiments, we have used the assets prices of 12 companies. Each data set has a minimum of 7 years of daily data and a maximum of 30 years. In order to simplify the study, we will be working with a 1-day horizon, that is, take a decision based on the forecasts of the assets variation for 1 day ahead. Moreover, we will be working exclusively with the closing prices of each trading session, that is, we assume trading decisions are to be made after the markets close.

The decision function for this application receives as input the forecast of the daily variation of the assets closing prices and returns a trading action. We will be using the following function in our experiments:

$$g : \mathbb{R} \rightarrow \mathcal{A} = \{hold, buy, sell\}$$
$$Y \mapsto \begin{cases} buy, & Y > 0.02 \\ sell, & Y < -0.02 \\ hold, & \text{other cases} \end{cases}.$$

This means we are assuming that any variation above 2% will be sufficient to cover the transaction costs and still obtain some profit. Concerning the data that will be used as predictors for the forecasting models (either forecasting the prices variation [$Y$] or directly the trading action [$A$]), we have used the price variations on recent days as well as some trading indicators, such as the annual volatility, the Welles Wilder style moving average (Wilder, 1978), the stop and reverse point indicator developed by J. Welles Wilder (Wilder, 1978), the usual moving average, and others. The goal of this selection of predictors is to provide the forecasting models with useful information on the recent dynamics of the assets prices.

Regarding the performance metrics, we will use to compare each approach; we will use two metrics that capture important properties of the economic results of the trading decisions made by the alternative models. More specifically, we will use the Sharpe Ratio as a measure of the risk (volatility) associated with the decisions and the percentage total return as a measure of the overall financial results of these actions. We will also consider the Macro versions of the Recall and Precision metrics for the buy and sell signals combined. This will allow us to analyze how many trading opportunities are being detected (Recall) and how accurate are the models regarding the prediction of every non-hold trading action (Precision), respectively. To make our experiments more realistic, we will consider a transaction cost of 2% for each buy or sell decision a model may originate.

At this stage, it is important to remark that the prediction tasks we are facing have some characteristics that make them particularly challenging. One of the main hurdles results from the fact that interesting events, from a trading perspective, are rare in financial markets. In effect, large movements of prices are not very frequent. This means that the data sets we will provide to the models have clearly imbalanced distributions of the target variables (both the numeric percentage variations and the trading actions). To make this imbalance problem harder, the situations that are more interesting from a trading perspective are rare in the data sets, which creates difficulties to most modeling techniques. In the next section, we will describe some of the measures we have taken to alleviate this problem. We have conducted a thorough analysis to test the impact of these measures on each approach. Evaluating these existing techniques in the context of financial forecasting problems is the second main contribution of the paper.

## 3.2 | The models

In this section, we describe all the model variants that will be used in the experimental comparisons. They were selected to ensure that both approaches have the same conditions for a fair comparison and that the results are not biased by some specific characteristics of one particular technique. Several variants for each family of models (SVM [support vectorial machines], Random Forests, etc.) were tested in order to make sure our conclusions were not biased. Tables 1 and 2 show all the model variants used in our experiments (nearly 182 model variants were tested). To facilitate the reproducibility of our results, we have used the free and open source implementations of these techniques available in the R software environment (R Core Team, 2014).

The predictive tasks we are facing have two main difficulties: (a) the fact that the distribution of the target variables is highly imbalanced, with the more relevant values being less frequent; and (b) the fact that there is an implicit ordering among the decisions. The first problem causes most modeling techniques to focus on cases (the most frequent) that are not relevant for the application goals. The second problem is specific to classification tasks as these algorithms do not distinguish among the different types of errors, whilst in our target application, confusing a buy decision with a hold decision is less serious than confusing it with a sell.

**TABLE 1** Regression models used for the experimental comparisons

| Model | Variants | R package |
|---|---|---|
| SVM | cost = {1,5,10}, $\varepsilon$ = {0.1,0.05,0.01}, tolerance = {0.001,0.005},kernel = linear | e1071 - Meyer, Dimitriadou, Hornik, Weingessel, and Leisch (2014) |
| SVM | cost = {1,10}, $\varepsilon$ = {0.1,0.05,0.01}, degree = {2,3,5},kernel = polynomial | e1071 - Meyer et al. (2014) |
| Random Forest | ntree = {500,750,1000,2000,3000},mtry = {4,5,6} | randomForest - Liaw and Wiener (2002) |
| Trees (pruned) | se = {0,0.5,1,1.5,2},cp = 0, minsplit = 6 | DMwR – Torgo (2010) |
| KNN | k = {1,3,5,7,11,15} | DMwR - Torgo (2010) |
| NNET | size = {2,4,6},decay = {0.05,0.1,0.15} | Nnet - Venables and Ripley (2002) |
| MARS | thresh = {0.001,0.0005,0.002}, degree = {1,2,3},minspan = {0,1} | Earth – Milborrow (2014) |
| AdaBoost | dist = {gaussian},n.trees = {10000,20000}, shrinkage = {0.001,0.01},interaction.depth = {1,2)} | Gbm – Ridgeway (2013) |

SVM = support vectorial machines; KNN = K-nearest neighbours; NNET = neural networks; MARS = multivariate adaptive regression spline.

**TABLE 2** Classification models used for the experimental comparisons

| Model | Variants | R package |
|---|---|---|
| SVM | cost = {1,3,7,10},kernel = linear tolerance = {0.001,0.005,0.0005,0.002} | e1071 - Meyer et al. (2014) |
| SVM | cost = {1,10}, $\varepsilon$ = {0.1,0.05}, degree = {2,3,4,5},kernel = polynomial | e1071 - Meyer et al. (2014) |
| Random Forest | ntree = {500,750,1000,2000,3000},mtry = {3,4,5} | randomForest - Liaw and Wiener (2002) |
| Trees (pruned) | se = {0,0.5,1,1.5,2},cp = 0, minsplit = 6 | DMwR – Torgo (2010) |
| KNN | k = {1,3,5,7,11,15} | DMwR – Torgo (2010) |
| NNET | size = {2,4,6},decay = {0.05,0.1,0.15} | Nnet - Venables and Ripley (2002) |
| AdaBoost | coeflearn = c('Breiman','Freund','Zhu'), mfinal = c(500,1000,2000) | Boosting - Ridgeway, Southworth, and RUnit (2013) |

SVM = support vectorial machines; KNN = K-nearest neighbours; NNET = neural networks; MARS = multivariate adaptive regression spline.

These two problems led us to consider several alternatives to our base modeling approaches described in Tables 1 and 2. For the first problem of imbalance, we have considered the hypothesis of using resampling to balance the distribution of the target variable before obtaining the models. In order to do that, we have used the SMOTE algorithm (Chawla, Bowyer, Hall, & Kegelmeyer, 2002). This method is well-known for classification models, consisting basically of oversampling the minority classes and under-sampling the majority ones. The goal is to modify the data set in order to ensure that each class is similarly represented. Regarding the regression tasks, we have used the work by Torgo, Branco, Ribeiro, and Pfahringer (2015), where a regression version of SMOTE was presented. Essentially, the concept is the same as in classification, using a method to try to balance the continuous distribution of the target variable by oversampling and under-sampling different ranges of its domain.

Regarding the second problem of the order among the classes, we have also considered a frequently used approach to handle this issue. Namely, we have used a cost–benefit matrix that allows to distinguish between the different types of classification errors. Using this matrix, and given a probabilistic classifier, we can predict for each test case the class that maximizes the utility instead of the class that has the highest probability.

We have used the following procedure to obtain the cost–benefit matrices for our tasks. Correctly predicted *buy*/*sell* signals have a positive benefit estimated as the average return of the *buy*/*sell* signals in the training set. On the other hand, in the case of incorrectly predicting a true *hold* signal as *buy* (or *sell*), we assign it minus the average return of the *buy* (or *sell*) signals. Basically, the benefit associated to correctly predicting one rare signal is entirely lost when the model suggests an investment when the correct action would be doing nothing. In the extreme case of confusing the *buy* and *sell* signals, the penalty will be minus the sum of the average return of each signal. Choosing such a high penalty for these cases will eventually change the model to be less likely to make this type of very dangerous mistakes. Considering the case of incorrectly predicting a true *sell* (or *buy*) signal as *hold*, we also charge for it but in a less severe way. Therefore, the average of the *sell* (or *buy*) signal is considered, but divided by two. This division was our way of "teaching" the model that it is preferable to miss an opportunity to earn money rather than making the investor lose money. Finally, correctly predicting a *hold* signal gives no penalty nor reward, because no money is either won or lost. Table 3 shows an example of such cost–benefit matrix that was obtained with the data from 1981-01-05 to 2000-10-13 of Apple.

We have thoroughly tested the hypothesis that using resampling before obtaining the models would boost the performance of the different models we have considered for our tasks (both classification and regression) and have also tested the hypothesis that using cost–benefit matrices to implement utility maximization would also improve the performance of the classification models. The results of testing these hypotheses will be presented in Section 4.

**TABLE 3** Example cost–benefit matrix for Apple shares

| | | Trues | | |
|---|---|---|---|---|
| | | S | H | B |
| Pred | S | 0.49 | −0.49 | −0.82 |
| | H | −0.24 | 0.00 | −0.17 |
| | B | −0.82 | −0.33 | 0.33 |

S = share; H = hold; B = buy.

### 3.3 | The experimental methodology

In this section, we present the experimental methodology used in our comparative experiments. Because of the temporal nature of the data sets, the usual cross-validation methodology should not be used to estimate the performance of a certain model. This procedure involves randomly reshuffling the data, which may lead to test cases that are "older" than the training cases, which would lead to unreliable estimates. In this context, we have used a Monte Carlo simulation method consisting of randomly selecting a series of $N$ points in time within the available data set. For each of these random dates, we use a certain consecutive past window as training set for obtaining the alternative models that are then tested/compared in a subsequent and consecutive test window. The Monte Carlo estimates are formed by the average scores obtained on the $N$ repetitions. In our experiments, we have used $N = 10$, 50% of the data as the size of the training window, and 25% of the data as size of the test sets.

With respect to testing the statistical significance of the observed differences between the estimated scores, we have used the recommendations of the work by Demšar (2006). More specifically, in situations where we are comparing $k$ alternative models on one specific task, we have used the Wilcoxon signed-rank test to check the significance of the differences. On the experiments where $k$ models are compared on $t$ tasks, we use the Friedman test followed by a post hoc Nemenyi test to check the significance of the difference between the average ranks of the $k$ models across the $t$ tasks.

## 4 | EXPERIMENTAL RESULTS

This section describes the results of our empirical studies. We have split these results in two main parts. The first has to do with testing the validity of the hypotheses described in Section 3.2 concerning the usage of resampling techniques and also the usage of cost–benefit matrices. The second part is the results of the final comparison between the two modeling approaches to our target decision problems.

### 4.1 | Addressing the particularities of the prediction tasks

As we have mentioned before, the prediction tasks we are facing have some particularities that turn them into particularly challenging problems. We have described two ways of trying to overcome these challenges. In this section, we check the validity of these hypotheses. Specifically, we test the advantages of: (a) using SMOTE on classification and regression models to overcome the problem of imbalanced distributions; and (b) using cost–benefit matrices on classification models to provide information on the different importance of the class values.

#### 4.1.1 | Hypothesis 1: Resampling the data sets

Our first hypothesis states that resampling the data sets with the goal of balancing the response variable will enhance the performance of both classification and regression models. In the experiments carried out to check the validity of this hypothesis, we have observed that the way the resampling affected each modeling approach was not significantly different. In this context, we will just present the results for the classification models as the conclusions with regression models were similar.

We start by comparing the top modeling variant obtained without using resampling against the best one using SMOTE, company by company, applying a Wilcoxon test in each case (for each modeling approach individually). In Figure 1 we analyze recall and precision for the buy and sell signals combined. The results were somewhat expected. Resampling methods balance more the distribution of the target, which means the models will have more examples of the rare cases that are interesting in this application. This has a positive impact on recall because the models end up forecasting more frequently these events because they are not so rare in the re-sampled training sets. However, due to the inherent difficulty of this financial forecasting tasks, this also involves a higher risk of making wrong predictions and thus the decrease in precision. Because we are in a trading context, we should prefer safer decisions rather than riskier ones. However, only after checking the financial results of these strategies we can confirm this. Nevertheless, we should reinforce that the issue of preferring high precision (less risk) over high recall is something specific to financial trading, and in other application domains, the conclusions on the usefulness of resampling could be different if the preference bias is different.

Figure 2 allows us to analyze the financial consequences of the resampling procedure. We can observe that in terms of Total Return and Sharpe Ratio, the models applied to data sets without any resampling achieved significantly better results and, in several cases, by a large margin. This means that pre-processing the data using resampling is leading to models that make more risky decisions and that these decisions are often wrong, leading to serious financial losses.

Considering all the plots at the same time, our results provide evidence that using SMOTE on the training data will make the models predict significantly more trading signals (Buy and Sell). This leads to higher recall but unfortunately also to much lower precision because these signals are frequently wrong, with serious financial consequences. However, we should not forget that the previous comparisons were carried out between the best overall variant of each type (with and without SMOTE). A different question is whether the same conclusions are valid if we consider each modeling technique individually. In this context, we will now check the resampling hypothesis per type of model instead of globally. We will group the variants according to the type of model and analyze the impact of using SMOTE per type of model.

The results of this new set of comparisons are shown in Figure 3. The first thing to notice is that the recall was significantly better every single time across all types of models where resampling was applied. On the other hand, the precision was worse almost every time, except with SVMs. So we again observe that the use of the SMOTE algorithm is making the models more capable of detecting the buy and sell (rare) classes at the cost of riskier decisions (except for SMVs). Concerning the financial metrics, we observe a clear

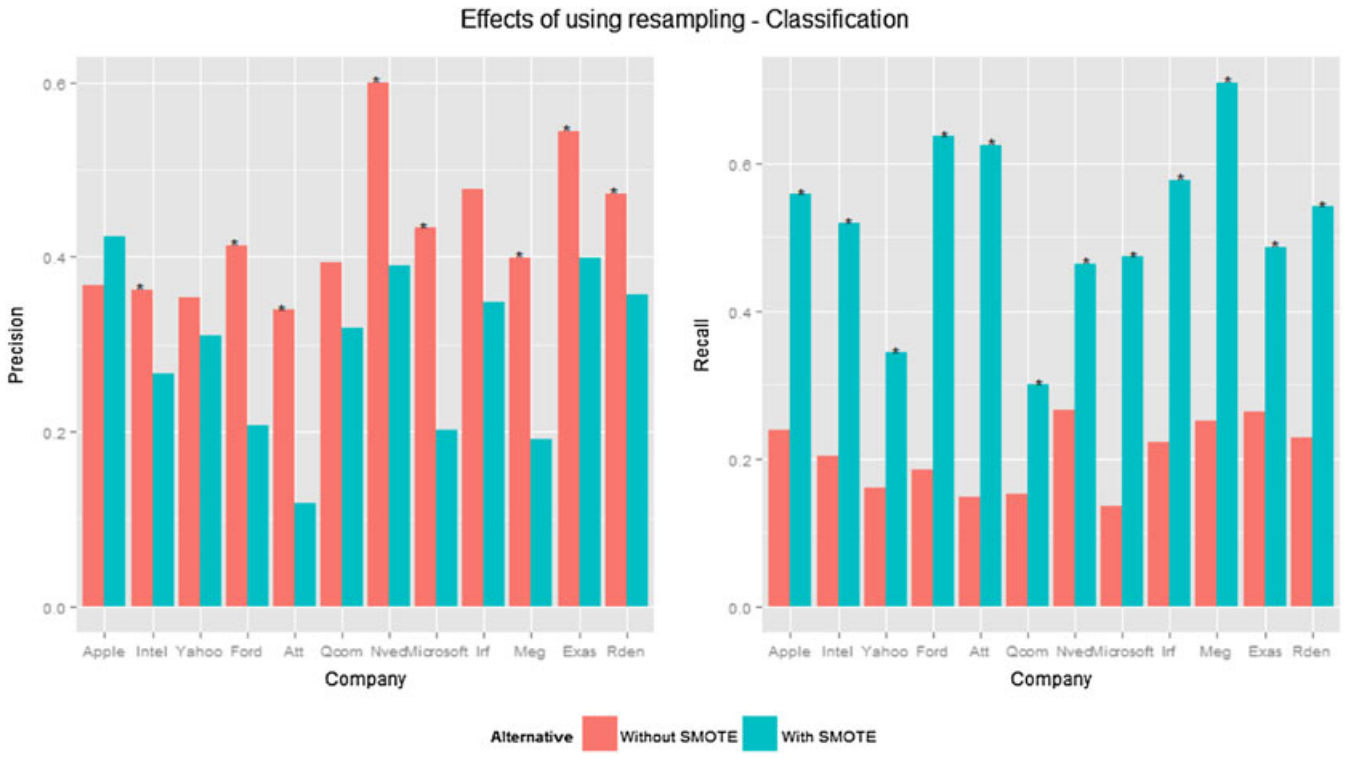


**FIGURE 1** Best classification variant without SMOTE against the best classification with SMOTE for the macro versions of Precision and Recall metrics (asterisks denote that the respective variant is significantly better, according to a Wilcoxon test with $\alpha = 0.05$)

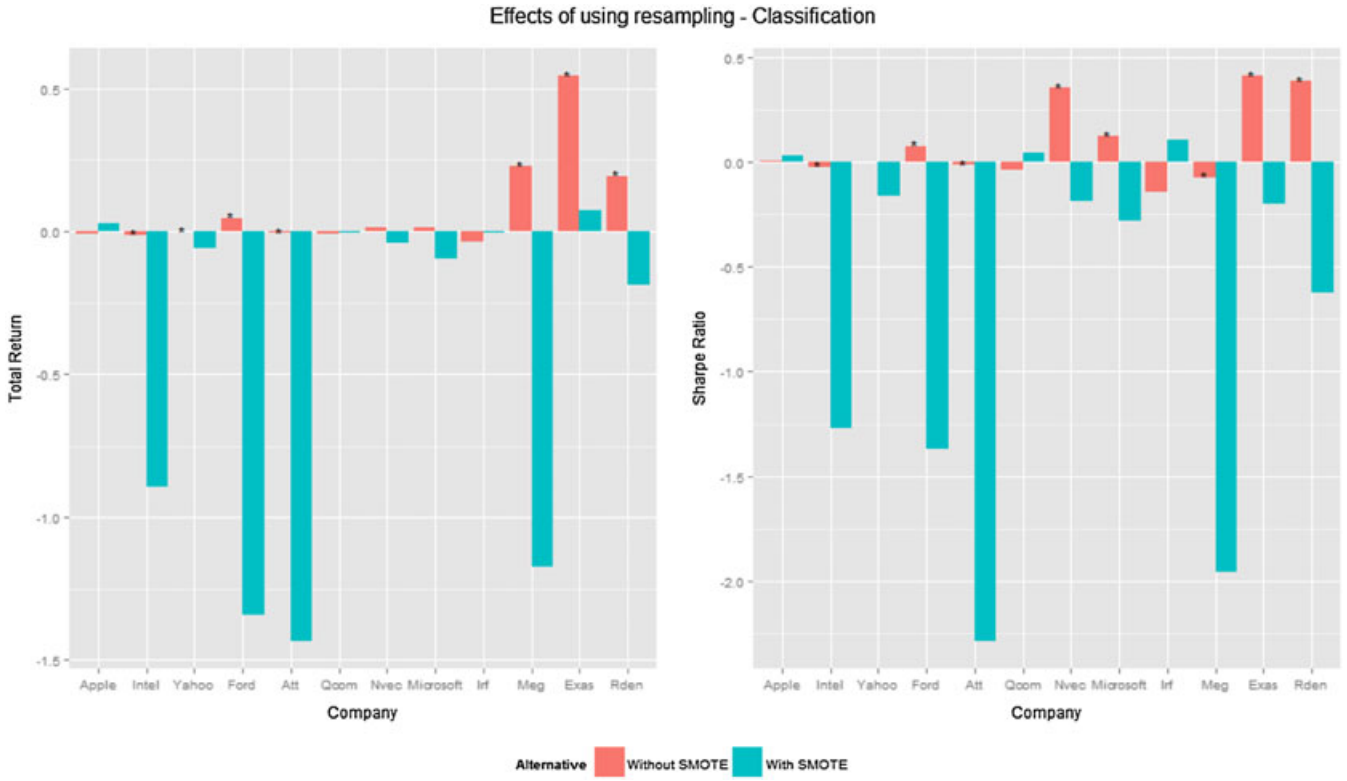


**FIGURE 2** Best classification variant without SMOTE against the best classification with SMOTE for the Total Return and Annualized Sharpe Ratio metrics (asterisks denote that the respective variant is significantly better, according to a Wilcoxon test with $\alpha = 0.05$)

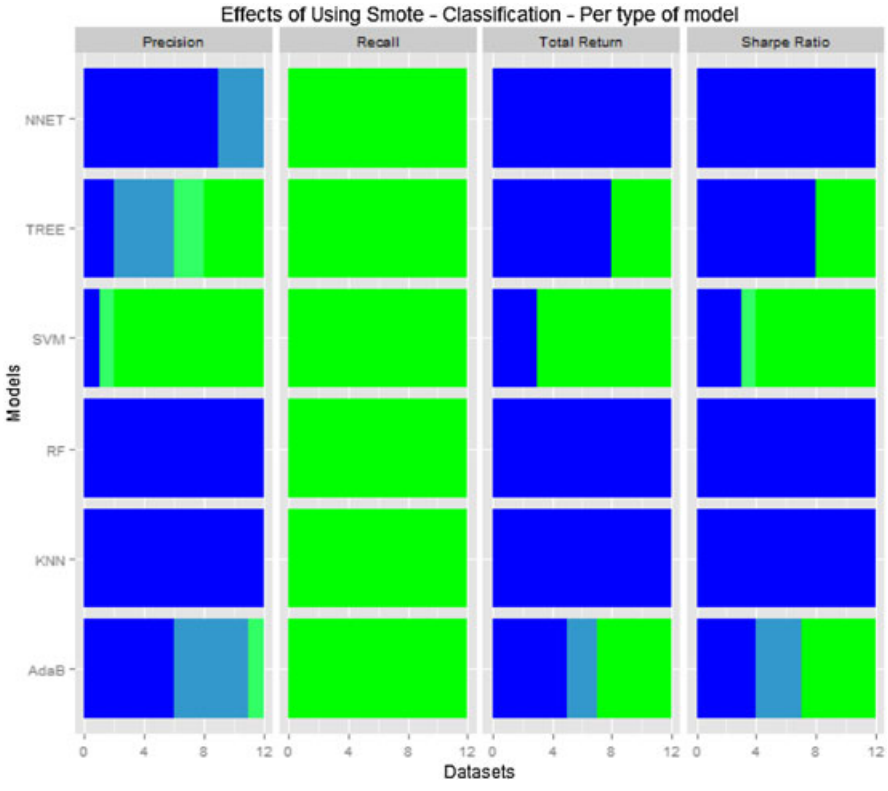


**FIGURE 3** Segmented by type of model and by metric, a Wilcoxon test is performed between the best model variant of each modeling tool (Classification without SMOTE versus Classification with SMOTE). Because there are 12 datasets in each segment, then there are 12 results per segment. Each bar shows the results for each segment, where each one of the four colors is associated to a type of win (significant/non-significant win without SMOTE - strong/light blue, non-significant/significant win with SMOTE - light/strong green). The length of each color describes the number of times that type of win occurred

advantage of the standard approach (i.e., no resampling applied), with SVM and AdaBoost being the only algorithms benefiting from the use of resampling on some tasks, with the advantage being more evident for SVMs.

In summary, we have collected sufficient evidence to conclude that resampling the data sets for both classification and regression models will have a negative impact on their performance in the context of financial forecasting tasks, namely when considering

trading-related evaluation metrics. Only the SVM, TREE, and AdaBoost algorithms have benefited from the resampling method on a small set of tasks. Overall, our conclusion is that this resampling strategy is not recommended in the context of forecasting for financial trading.

#### 4.1.2 | Hypothesis 2: adding cost–benefit matrices

The second hypothesis we have put forward was that the use of cost–benefit matrices would boost the performance of the classification models, because the information on the implicit ordering among the classes would be passed to the models, allowing them to avoid more costly errors (e.g., confusing a buy decision with a sell decision).

In order to test this hypothesis, we will follow the same methodology used for the resampling hypothesis. Firstly, the top modeling variant of each alternative (with and without cost–benefit matrices) will be compared for each data set. The results regarding precision and recall are shown in Figure 4.

These results are a bit inconclusive. In terms of recall the use of these matrices led to better results, as we have five significant wins against only one significant loss of the approach using the matrices. On the other hand, in terms of precision, we observed three significant wins of the approach without cost–benefit matrices.

Figure 5 shows the results of this same experiment in terms of the financial metrics. The most evident observation is the fact that not a single statistically significant difference was achieved by either alternative. However, both in terms of the Total Return and of the Annualized Sharpe Ratio, the approaches using cost–benefit matrices obtained more wins. This is an interesting result, suggesting that the use of these matrices may be beneficial for the classification models in the context of financial trading based on prediction models.

Let us now check if these results hold across each different type of learning algorithm. Figure 6 shows the comparison between the top modeling variant of each type of model. Even though there is a slightly higher abundance of green (suggesting that the usage of cost–benefit matrices may be beneficial), these results are quite even. Each type of model is influenced in a different way by the cost–benefit matrices. Although NNET (neural networks), TREE (decision tree models) and SVM are able to take advantage of the information on the matrices, the conclusions for the other models are not so clear. These observations seem to indicate that the potential advantage of the usage of cost–benefit matrices is algorithm-dependent, with some techniques being able to capitalize on this extra information while others do not. As future research agenda it, should be interesting to understand if there are some theoretical properties of the algorithms that may be causing this differentiated behavior. A possible explanation has to do with the quality of probability estimates. In effect, the decisions using cost–benefit matrices depend on the estimated probabilities of each class. If these estimates are wrong or unreliable, this may lead to unreliable decisions. Still, this explanation needs to be confirmed in practice.

In summary, contrary to the first hypothesis involving resampling, the conclusions for the second hypothesis regarding the advantages of

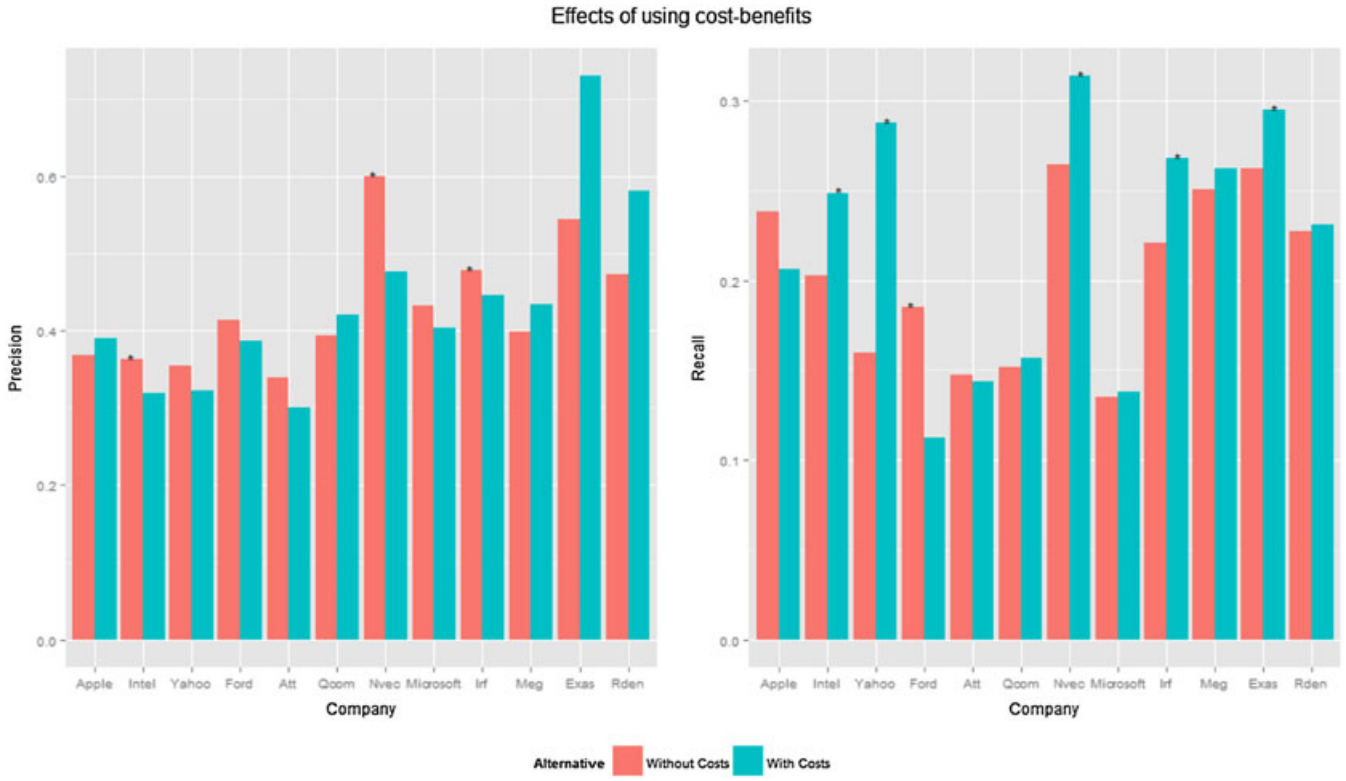


**FIGURE 4** Best classification variant without costs against the best classification with costs for the macro versions of Precision and Recall metrics (asterisks denote that the respective variant is significantly better, according to a Wilcoxon test with $\alpha = 0.05$)

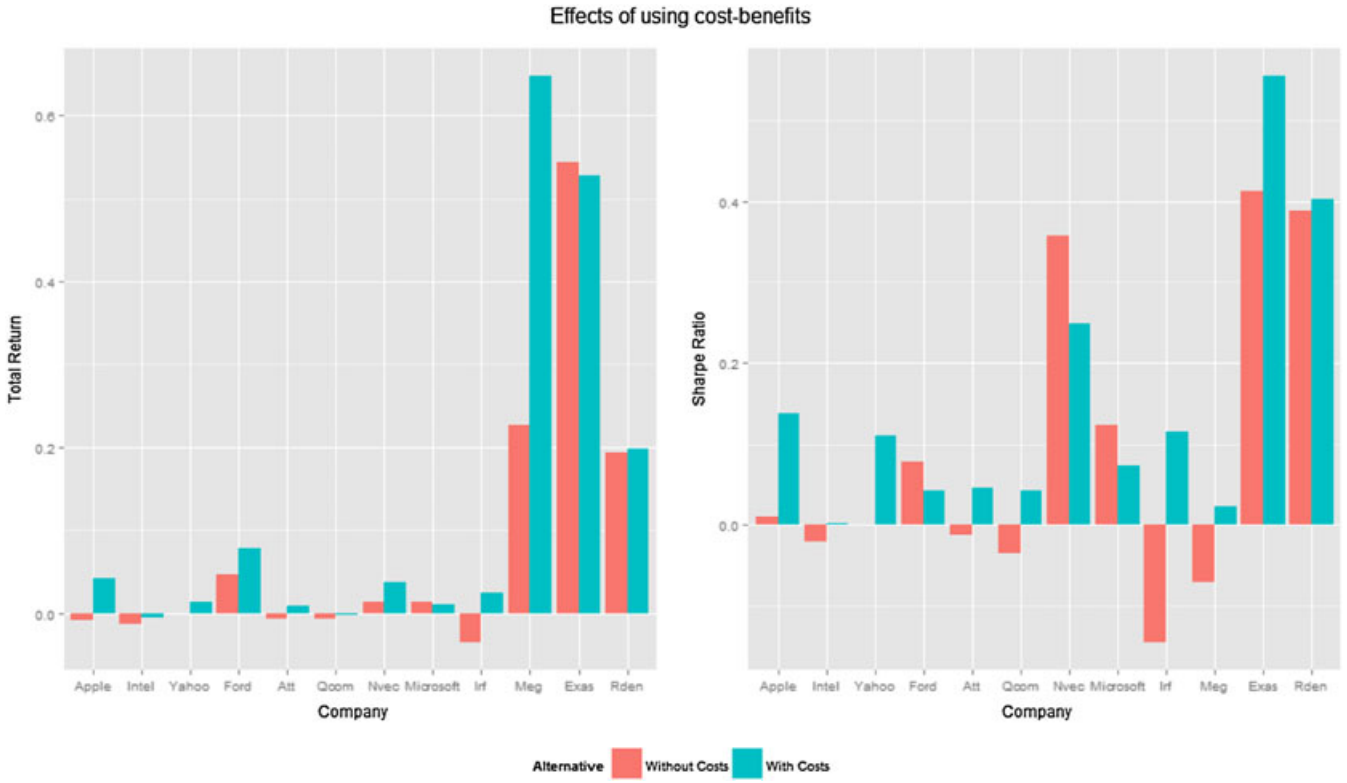


**FIGURE 5** Best classification variant without costs against the best classification with costs for the Total Return and Annualized Sharpe Ratio metrics (asterisks denote that the respective variant is significantly better, according to a Wilcoxon test with $\alpha = 0.05$)

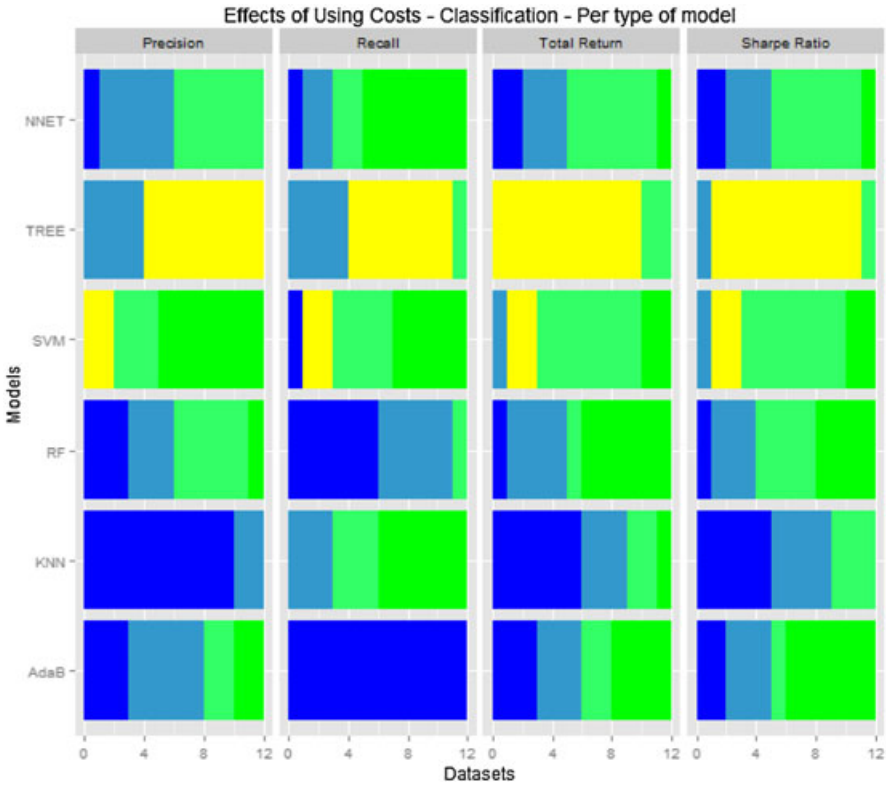


**FIGURE 6** Segmented by type of model and by metric, a Wilcoxon test is performed between the best model variant of each modeling tool (Classification without costs versus Classification with costs). Because there are 12 datasets in each segment, then there are 12 results per segment. Each bar shows the results for each segment, where each one of the five colors is associated to a type of win (significant/non-significant win without costs - strong/light blue, draw - yellow, non-significant/significant win with costs - light/strong green). The length of each color describes the number of times that type of win occurred

cost–benefit matrices are that there is some potential for this alternative way of addressing the classification tasks. Although we have not observed an overwhelming advantage of this usage, we have found that for some modeling algorithms, these matrices provide a clear boost in terms of their results (both in terms of Precision/Recall and financially-oriented metrics).

### 4.2 | Comparison of classification and regression modeling approaches

This section presents the results of the experimental comparisons between the two general approaches to making trading decisions based on forecasting models. In our experiments, we have considered 76 classification models. For each of these models, we have also tried the version with resampling and the version with cost–benefit matrices, totaling 76 × 3 = 228 different classification variants. In terms of regression, we have a slightly larger set of 97 base models that were then tried with and without resampling, for a total of 97 × 2 = 194 variants. All these variants were compared on the data sets of the 12 companies described in Section 3.1 using the methodology described in Section 3.3.

We have divided our experimental analysis in two main parts. In the first one, for each company and for each metric, we have compared the best regression and classification variant using a Wilcoxon singed-rank statistical test with a significance level of 0.05 to check if we can reject the null hypothesis that there is no significant difference between the best classification and regression variants. This leads to 12 statistical tests for each metric (one test for each company), where the models compared for each company are not necessarily the same. The motivation of this first part is to compare the best classification variant against the best regression variant for each company and metric. Figure 7 shows the results of this comparison for the Total Return and Sharpe Ratio financial evaluation metrics. The results on these figures are somewhat correlated. In effect, whenever we have found a significant difference in terms of Total Return, the same also happened in terms of Sharpe Ratio. Regarding the left graph (Total Return), we have one significant win for each approach and 6 against four non-ignificant wins for classification and regression, respectively. With respect to the right graph (Sharpe Ratio), we can observe a slight advantage of the classification approach, with one more significant win and eight vs one non-significant wins. Overall, we have observed a very slight advantage of the best classification approach against the best regression variant.

From an economical perspective, some results are contradictory. For instance, there is a very high level of Total Return for the Meg company (above 60% return), but the best Sharpe Ratio was very low. This means that the best model for the first metric was taking enormous amounts of risk and that the high level of return achieved was probably due to pure luck. On the other hand, there are some high values for the Total Return accompanied by high levels of Sharpe Ratio, such as for the Exas company. This strongly suggests that the models could actually provide some profit with low risk, thus indicating that the model actually predicted meaningful signals. Given the high variability of the results across companies, taking conclusions solely based on the analysis of the best variant per model and per metric may lead to wrong results. This establishes the motivation for the second part of our experiments.

In this second part of our experiments, instead of grouping by metric and company, we will just group by metric and study the average rank of each model across all the companies (top five of each approach are considered). With the use of the Friedman test followed by the post hoc Nemenyi test, we check whether there are statistically

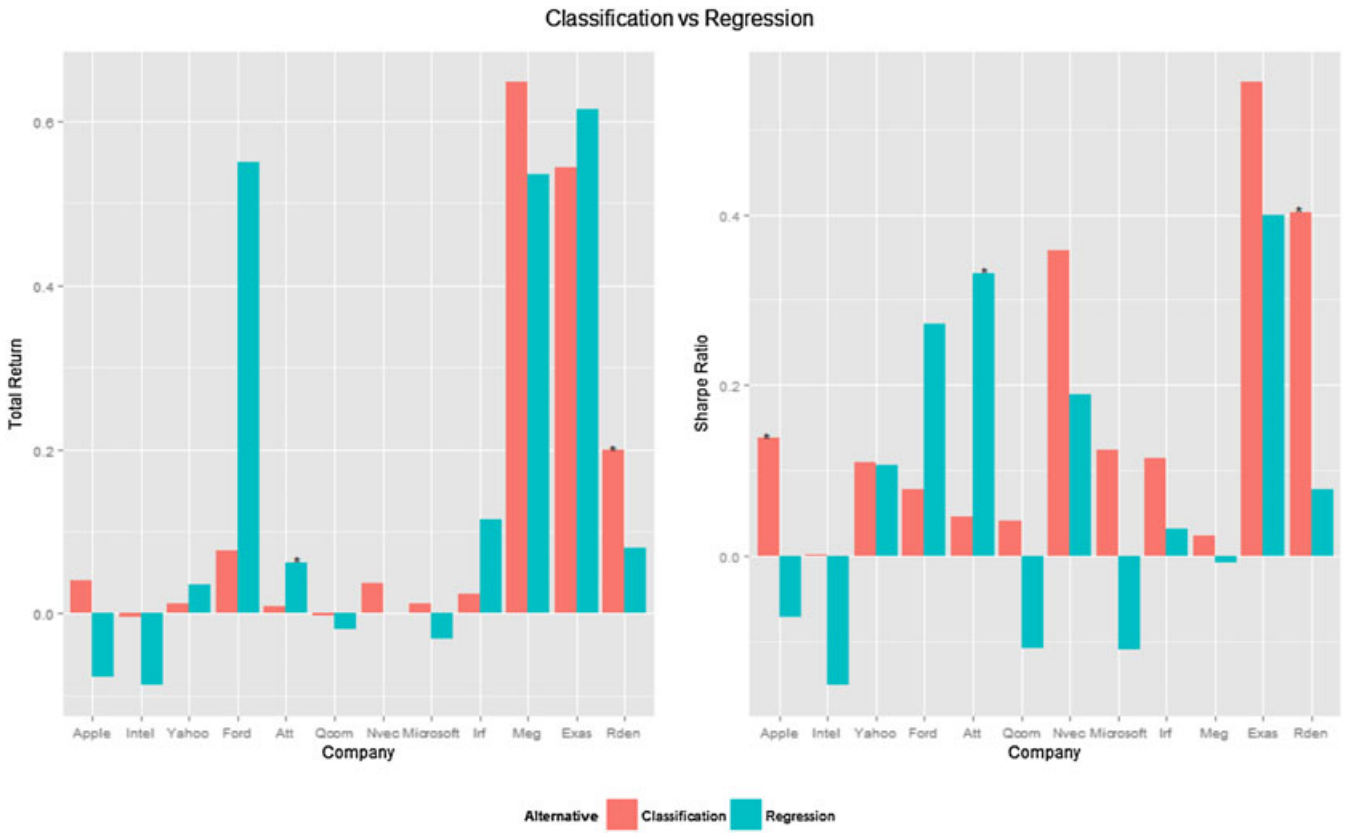


**FIGURE 7** Best classification variant against the best regression one for the Total Return and Sharpe Ratio metrics (asterisks denote that the respective variant is significantly better, according to a Wilcoxon test with $\alpha = 0.05$)

significant differences among these rankings. This way, if a model obtains a very good result for one company but poor for all the others (meaning that it was lucky in that specific company), its average ranking will be low allowing the top average rankings to be populated by the true top models that perform well across most companies.

Figure 8 summarizes the results in terms of Total Return and Sharpe Ratio. In either case, because we could not reject the Friedman null hypothesis, the post hoc Nemenyi test was not performed. This means that we cannot say with 95% confidence that there is some significant difference in terms of the average rank of the models for both the Total Return and the Sharpe Ratio between these two modeling approaches. Nevertheless, there are some observations to remark. Regarding Total Return, the model with the best average ranking is a classification model using cost–benefit matrices. All the remaining classification variants are in their original form (without using cost–benefit matrices) and occupying mostly the last positions in terms of average rankings. Moreover, not a single variant obtained with SMOTE appears in this top five for each approach, which means that we confirm that resampling does not seem to pay off for this type of applications due to the economic costs of making more risky decisions. Furthermore, another very interesting remark is that all the top models are using SVMs as the base learning algorithm. Overall, we cannot say that any of the two approaches (forecasting directly the trading actions using classification models or forecasting the price returns using regression) is better than the other regarding the Total Return.

The second part of Figure 8 shows the results of the same experiment in terms of Sharpe Ratio, that is, the risk exposure of the alternatives. The conclusions are quite similar to the Total Return metric. Once again, no significant differences were observed. Still, one should note that the first five places are dominated by the classification approaches. The best variant for the Total Return is also the best variant for the Sharpe Ratio, which makes this variant unarguably the best one of our study when considering the 12 different companies. Hence, ultimately, we can state that the most solid model belongs to the classification approach using an SVM with cost–benefit matrices, because it obtained the highest returns with lowest associated risk. Finally, unlike the results for Total Return, in this case, we observe other learning algorithms appearing in the top five best results.

In conclusion, we cannot state that one approach performs definitely better than the other in the context of financial trading decisions. The scientific community typically puts more effort into the regression models, but this study strongly suggests that both have at least the same potential. Actually, the most consistent model we could obtain belongs to the classification approach. Another interesting conclusion is that of a considerably large set of different types of models, SVMs achieved better results both when considering classification or regression tasks.

## 5 | DISCUSSION

In this paper, we have studied the question of classification versus regression on decision making problems based on forecasts of a numeric variable. Our study was focused on financial trading decisions, so the obvious question is as follows: Can these results be generalized to other domains/tasks with similar structure?

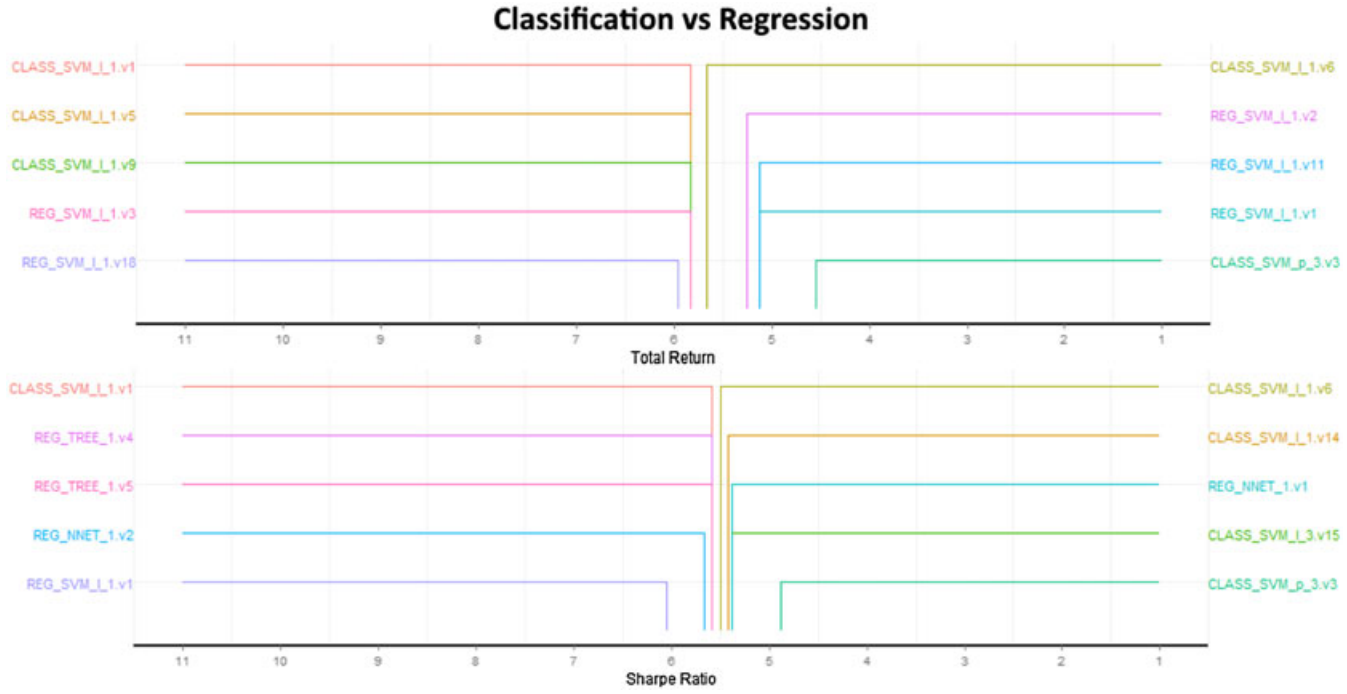


**FIGURE 8** The top five average ranking model variants of both modeling approaches (Classification and Regression) are forming a new set of variants, and their average rankings are recalculated. Each model is thus given an average ranking (x axis). The presence of at least one black line implies that the Friedmans null hypothesis of all averages being equal was rejected. In that case, every pair of model variants not connected by any black line is considered to have their average rankings significantly different according to a Nemenyi statistical test

This question was addressed in Baia (2015). The setup used in that work was rather similar to the one presented here, with two main differences: (a) the tasks were associated to several distinct contexts (while the current paper focus only on trading tasks); and (b) each task was considered with a different number of possible decisions to be made. While the second point allowed us to evaluate if one modeling approach could gain some specific advantage over the other depending on the number of values of the decision variable, the first point let us check if whatever conclusion we reached regarding trading problems would also apply in a more generic context. Moreover, the tasks considered in Baia (2015) were all non-temporal, stressing out even more the differences to the trading problem.

The main conclusions of the study carried out in Baia (2015) were that, overall, the classification modeling approach can outperform more frequently the approach based on regression tools. Whether the user is willing to make an extensive search for the optimal parameters to model a certain task or not, the results point in the same direction. The methods based on the classification approach tend to be better. The only setup where the regression approach was more competitive was on tasks with a high number of classes/decisions and where the user is not merely interested in the accuracy of the decisions and wants to consider different grades of severity of the decision errors.

Contrary to what was observed in Baia (2015), in this paper, we have gathered enough evidence to state that there is no statistically significant difference between both approaches in the context of the trading decision problem. The trading problem has some characteristics that make it quite different from the generic tasks studied in Baia (2015), which may explain the different conclusions. Namely, the tasks addressed in the current paper are based on data that is ordered by time (time series data), and the frequency of decisions is rather imbalanced, with the more important decisions being rare. These characteristics raise significant challenges to modeling tools, and even with the use of techniques developed to address these issues, we were not able to obtain conclusive results. In summary, the conclusions of the current paper seem to be specific to the financial trading setting. For more general tasks, existing evidence (Baia, 2015) seems to indicate that there is some advantage on using the classification approach. Future work should try to explain the reason for this different conclusion, namely if it is something specific to the domain or to some of its characteristics (time-dependent data and imbalanced decisions).

## 6 | RELATED WORK

To the best of our knowledge, this is the first research work to directly compare the regression and classification approaches to financial trading. Existing work in this area essentially uses one of the approaches and compares different variants of it on some concrete financial data sets. Even outside of the financial trading domain, we were not able to find some comparison between these two plausible approaches to decision making based on numeric forecasts. Still, as we have mentioned in the introduction, we think this is a common and relevant setup for many application domains.

Nevertheless, there are some concepts that are strongly related with the problem of making decisions based on numeric forecasts and thus we provide here a short review of some of the main works in these areas. In this paper, the ultimate goal is to predict an action/decision. However, in many application domains, there are decisions that are more important than others, and there may exist information on some costs and benefits associated with each decision. In this context, all area of cost-sensitive learning (Elkan, 2001) is strongly related with our target applications. Another related problem is that of imbalanced distributions of the target variable in the context of prediction models. Both these two problems are present in financial trading problems. Branco, Torgo, and Ribeiro (2016) present a survey of existing techniques for handling these situations of imbalanced target variables. Although most of the existing work considers classification tasks (nominal target variables), this work also describes methods designed to handle similar problems within regression tasks (numeric target variables).

In terms of the use of regression models for trading systems, neural networks models have shown to be a promising tool for forecasting time series (namely the prices of some assets). However, they typically require a large effort in terms of model tuning and can be computationally demanding. Genay (1999) has observed that a simple feed-forward network fails to statistically outperform the random walk model when the input variables are just the past returns. However, with the simple addition of a moving average, it can statistically outperform this baseline. Ghazali, Hussain, and Liatsis (2011); Shin and Ghosh (1995) and Sermpinis, Dunis, Laws, and Stasinakis (2012) have proposed variants and generalizations of neural networks that present decent improvements over the standard versions of these models. Most research seems to indicate that these standard versions of the neural networks models may present poor results, yet some small variations in terms of the structure of the model may greatly improve their performance.

Another popular modeling technique in this area is k-nearest neighbours (KNN). Genay (1999) has observed that the regression KNN model statistically outperforms the random walk model by merely using the past return as predictors, unlike the feed-forward neural network. Lee, Wei, Cheng, and Yang (2012) have used the nearest-neighbour-based approach for churn predictions. Even though this problem is different from financial trading, there are some similarities. Detecting an uncommon event the soonest possible can be seen as highly relevant in trading, that is, detecting a buy or sell signal the earliest possible to obtain the maximum possible profit. Support Vector Machines (SVM) and Multivariate Adaptive Regression Splines (MARS) have been the subject of a study by Kao, Chiu, Lu, and Chang (2013). The author also tested these models incorporated with wavelets, where some interesting results were obtained. Furthermore, Lu et al. (2009) has studied the combination of using independent component analysis to reduce the noise and randomness of the data before applying standard SVM models and has observed a slight improvement of the results.

In terms of using classification models in the context of financial trading, there are not some many works. Chang, Fan, and Liu (2009) have studied the combination of piecewise linear models with back-propagation artificial classification neural networks PLR-BPN. The experimental results were interesting in terms of the amount of profit

obtained. However, Luo and Chen (2013) have seen that PLR-BPN is outperformed by the combination of PLR with the well-known Support Vector Machine model. Ma et al. (2012) have used cost matrices with back-propagation neural networks. In several tasks, an increase of the utility score of a model came at the cost of a decrease in the accuracy level. However, this accuracy decrease may not be relevant for financial trading, where the main goal is to avoid serious errors like forecasting a buy signal when we should sell, as this type of errors may have very serious economic impact. Teixeira and de Oliveira (2010) have studied the classification KNN model and also the combination of this model with indicators such as the RSI filter, stop-gain and stop-loss criteria. All the tested models outperformed the used benchmark, particularly the models combined with the above indicators that obtained an overall better performance.

Atsalakis and Valavanis (2009) contain a very detailed state of art review on stock market forecasting techniques. For each referred work, the authors list the used data sets, the chosen input variables and, most importantly, a summary of all the used modeling techniques as well as which models were compared against each other. Information regarding the usage of pre-processing techniques and the training method was also given.

In summary, although our review of the related literature has not found any work with objectives similar to the current paper, we were able to observe that the most frequent approach to financial trading is based on regression approaches. Our work has shown that classification approaches should be given more importance by the research community in this area.

## 7 | CONCLUSIONS

This paper presents a comparative study of two different approaches to financial trading decisions based on forecasting models. The first, and more conventional approach, uses regression tools to forecast the future evolution of prices and then uses some decision rule based on these predictions, to choose the trading action. The second approach tries to directly forecast the "correct" trading decision. Our study is a specific instance of the more general problem of making decisions based on numerical forecasts. In this paper we have focused on financial trading decisions because this is a domain that requires specific trade-offs in terms of economic results. This means that our conclusions are specific to this area and it remains an open question for future research whether the same conclusions can be drawn in other application domains where the same two approaches to decision making are plausible. Still, our initial experiments (Baia, 2015) with other types of domains provide some evidence for the specificity of financial trading decision making that is addressed in the current paper.

Overall, the main conclusion of the study we have described in this paper is that, for this specific application domain, there seems to be no statistically significant difference between these two approaches to decision making. Given the large set of classification and regression models that were considered, as well as the different data sets involved in our study, we claim that this conclusion is supported by significant experimental evidence.

Financial forecasting has some particularities that are challenging to most modeling techniques. In our study we have considered the hypothesis of using some existing techniques that were developed to address this type of challenges. Another contribution of this paper involves testing the applicability of these solutions in the context of financial forecasting. Regarding the problem of the imbalance of the distribution of the target variable we have considered the application of resampling strategies both for regression and classification models. This technique has been applied with success to many application domains. Our experiments have shown that although resampling increases the ability of the models to generate trading signals (thus increasing their recall), it also brings a significant amount of financial risk as several of these signals are wrong, frequently leading to catastrophic financial results. This means that our study clearly indicates that resampling is not recommended in the context of financial forecasting due to this increase in the risk exposure. The other standard technique we have considered in our study was the use of cost–benefit matrices as a means to make the classification models aware of the different costs of the classification errors. In this case, our study collected sufficient evidence to conclude that this method is promising, although not all modeling techniques were able to capitalize on the extra information provided by these matrices.

## ACKNOWLEDGEMENTS

This work is financed by the European Regional Development Fund (ERDF) through the Operational Programme for Competitiveness and Internationalisation - COMPETE 2020 Programme within project POCI-01-0145-FEDER-006961 and by the North Portugal Regional Operational Programme (ON.2 - O Novo Norte), under the National Strategic Reference Framework (NSRF), through the European Regional Development Fund (ERDF), and by national funds, through the Portuguese funding agency (FCT) within Project NORTE-07-0124-FEDER-000059. Part of the work of Luís Torgo was supported by a sabbatical scholarship (SFRH/BSAB/113896/2015) from the Portuguese funding agency (FCT).

## REFERENCES

Atsalakis, G. S., & Valavanis, K. P. (2009). Surveying stock market forecasting techniques part II: Soft computing methods. *Expert Systems with Applications*, *36*(3, Part 2), 5932–5941.

Baia, L. (2015). Actionable forecasting and activity monitoring: Applications to financial trading.

Branco, P., Torgo, L., & Ribeiro, R. (2016). A survey of predictive modeling on imbalanced domains. *ACM Computing Surveys (to appear)*.

Chang, P.-C., Fan, C.-Y., & Liu, C.-H. (2009). Integrating a piecewise linear representation method and a neural network model for stock trading points prediction. *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, *39*(1), 80–92 .cited By 39

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, *16*(1), 321–357.

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

Elkan, C. (2001). The foundations of cost-sensitive learning. In *IJCAI'01: Proc. of 17th Int. Joint Conf. of Artificial Intelligence*, volume 1, pp. 973–978. Morgan Kaufmann Publishers.

Genay, R. (1999). Linear, non-linear and essential foreign exchange rate prediction with simple technical trading rules. *Journal of International Economics*, *47*(1), 91–107.

Ghazali, R., Hussain, A. J., & Liatsis, P. (2011). Dynamic ridge polynomial neural network: Forecasting the univariate non-stationary and stationary trading signals. *Expert Systems with Applications*, *38*(4), 3765–3776.

Hellstrom, T. (1999). Data snooping in the stock market. *Theory of Stochastic Processes*, *(21, 1999b)*, pp. 33–50.

Kao, L.-J., Chiu, C.-C., Lu, C.-J., & Chang, C.-H. (2013). A hybrid approach by integrating wavelet-based feature extraction with {MARS} and {SVR} for stock index forecasting. *Decision Support Systems*, *54*(3), 1228–1244.

Lee, Y.-H., Wei, C.-P., Cheng, T.-H., & Yang, C.-T. (2012). Nearest-neighbor-based approach to time-series classification. *Decision Support Systems*, *53*(1), 207–217.

Liaw, A., & Wiener, M. (2002). Classification and regression by random forest.

Lu, C.-J., Lee, T.-S., & Chiu, C.-C. (2009). Financial time series forecasting using independent component analysis and support vector regression. *Decision Support Systems*, *47*(2), 115–125 .cited By 112

Luo, L., & Chen, X. (2013). Integrating piecewise linear representation and weighted support vector machine for stock trading signal prediction. *Applied Soft Computing*, *13*(2), 806–816.

Ma, G.-Z., Song, E., Hung, C.-C., Su, L., & Huang, D.-S. (2012). Multiple costs based decision making with back-propagation neural networks. *Decision Support Systems*, *52*(3), 657–663.

Meyer, D., Dimitriadou, E., Hornik, K., Weingessel, A., & Leisch, F. (2014). e1071: Misc Functions of the Department of Statistics (e1071), TU Wien. R package version 1.6–4.

Milborrow, S. (2014). Earth: Multivariate adaptive regression spline models. R package version 3.2–7.

R Core Team (2014). *R: A language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing.

Ridgeway, G. (2013). GBM: Generalized Boosted Regression Models. R package version 2.1.

Ridgeway, G., Southworth, M. H., & RUnit, S. (2013). Package gbm.

Sermpinis, G., Dunis, C., Laws, J., & Stasinakis, C. (2012). Forecasting and trading the eur/usd exchange rate with stochastic neural network combination and time-varying leverage. *Decision Support Systems*, *54*(1), 316–329.

Shin, Y., & Ghosh, J. (1995). Ridge polynomial networks. *IEEE Transactions on Neural Networks*, *6*(3), 610–622 .cited By 64

Teixeira, L. A., & de Oliveira, A. L. I. (2010). A method for automatic stock trading combining technical analysis and nearest neighbor classification. *Expert Systems with Applications*, *37*(10), 6885–6890.

Torgo, L. (2010). *Data Mining with R, learning with case studies*. London, United Kingdom.

Torgo, L., Branco, P., Ribeiro, R. P., & Pfahringer, B. (2015). Resampling strategies for regression. *Expert Systems*, *32*(3), 465–476.

Venables, W. N., & Ripley, B. D. (2002). *Modern applied statistics with S* (4th ed. ISBN 0-387-95457-0). London, United Kingdom.

Wilder, J. (1978). *New concepts in technical trading systems*. Kingston, New York: Trend Research.

**How to cite this article:** Baía L, Torgo L. A comparative study of approaches to forecast the correct trading actions. *Expert Systems*. 2017;34: e12169. https://doi.org/10.1111/exsy.12169

## AUTHOR BIOGRAPHIES

Luis Baia has a master degree in Applied Mathematics and a Bachelor in Pure Mathematics. After some months researching in Machine Learning at LIAAD, he joined a company called “Farfetch” as a Data Scientist.

Luis Torgo is an associate professor in the Department of Computer Science at the University of Porto in Portugal. An active researcher in machine learning and data mining for more than 20 years, Dr. Torgo is also a researcher in the Laboratory of Artificial Intelligence and Data Analysis (LIAAD) of INESC Porto LA.