

FACULTY OF SCIENCE OF UNIVERSITY OF PORTO

Actionable Forecasting and Activity Monitoring: applications to financial trading

Luís Baía



Masters in Engineering Mathematics

Supervisor: Luís Torgo

August 26, 2015

Actionable Forecasting and Activity Monitoring: applications to financial trading

Luís Baía

Masters in Engineering Mathematics

August 26, 2015

Abstract

The thesis addresses a particular class of decision making problems. We name these tasks *Actionable Forecasting* problems. The main distinguishing feature of this type of decision problems is the fact that decisions are to be taken based on predictions of a numeric variable. Examples of such tasks include for instance the medical diagnosis of a patient based on predictions of some numerical indicator, or deciding which trading action should be taken depending on the prediction of the future evolution of the market prices. We study and compare two different alternative ways of addressing these decision problems: (i) using standard regression models to forecast the numeric variable and on a second step transform these numeric predictions into a decision according to some pre-defined and deterministic decision rules; and (ii) use models that directly forecast the right decision using classification models thus ignoring the intermediate numeric forecasting task. The objective of this 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. We also consider some potential limitations of each alternative, where we consider solutions such as the usage of cost-benefit matrices as well as re-sampling algorithms.

We carry out two major studies to compare both alternatives to solve actionable forecasting tasks: (i) one involving a large set of generic and non-temporal tasks; (ii) and a second involving financial trading problems that use price time series with the goal of deciding whether to invest or not in the market with short and long positions.

Decision making in the context of financial trading is a very relevant problem with very high economic impact. We argue that these tasks can be solved as a special instance of actionable forecasting problems.

We have gathered enough experimental evidence to support the conclusion that classification models may be preferable to model the generic tasks, mainly when used together with cost-benefit matrices. We have also observed some differences according to the number of possible decisions per task. The higher the number of possible decisions/actions to make, the higher will be the advantage of the classification models over the alternative of using regression models. With respect to the specific tasks of financial trading, both modelling alternatives revealed similar potential. The usage of re-sampling on such tasks brings too much risk to the models while using cost-benefit matrices on the classification models was beneficial once again.

The last topic addressed in this thesis was the evaluation of financial trading systems. Based on the theoretical framework of activity monitoring, we have proposed a formalisation of financial trading as an instance of these data mining tasks. Using this formalisation we have described algorithms that allow to obtain the ideal timings for holding market positions given some trading preference criteria. These ideal timings can be used as an optimal benchmark against which real trading records can be compared to. The main advantage of this evaluation framework is its adaptability to the investor's preference bi-

ases and also the interpretability of the evaluation outcomes. We present the evaluation framework and test it using trading records of real investors.

Acknowledgements

I would like to thank my parents and family for guiding and providing me with the wisdom to successfully choose, start and finalise this path. Whatever my successful accomplishments may be, they will also be my family's.

When it comes to my academic environment, I wish to express my sincere gratitude to my supervisor, Luís Torgo, not only for his great knowledge and expertise, but also for his constant support and guidance. I have significantly improved my scientific knowledge and taken the first steps towards the path of becoming an enthusiastic researcher. Certainly, a wonderful and enlightening experience.

Last, but not least, a heartfelt thank you to the close friends that endured and grew with me during the bachelor's and the master's, as well as the friends that have kept in touch through the past few years.

A final and kind word for my beloved partner who has been present through all my academic career and supported all my efforts. Thank you my dear.

Luís Baía

Contents

1	Introduction	1
1.1	Literature Revision	2
1.1.1	Actionable Forecasting	2
1.1.2	Evaluation of Trading Performance	4
1.2	Thesis Organisation	6
2	Actionable Forecasting	7
2.1	Problem Formalization	7
2.2	Proposed approaches	8
2.2.1	The Classification Approach: A limitation	9
2.3	Material and Methods	10
2.3.1	The Tasks	10
2.3.2	Evaluation Metrics	12
2.3.3	The Models	14
2.4	The Experimental Methodology	14
2.5	Analysis of the usage of cost-benefit matrices	17
2.5.1	Wilcoxon test: Best per metric and per data set	18
2.5.2	Wilcoxon test: Best per metric, per data set and per type of model	23
2.5.3	Post-hoc Nemenyi test: Average Ranks	27
2.5.4	Conclusions	31
2.6	Comparison of Regression and Classification modelling approaches	33
2.6.1	Wilcoxon test: Best per metric and per data set	34
2.6.2	Post-hoc Nemenyi test: Average Ranks	35
2.7	Conclusions	43
3	An Application to Financial Trading	45
3.1	Material and Methods	46
3.1.1	The Task	46
3.1.2	Evaluation Metrics	47
3.1.3	The Models	48
3.2	The Experimental Methodology	49
3.3	Hypothesis testing	49
3.3.1	Hypothesis 1: Re-sampling the data sets - Classification	50
3.3.2	Hypothesis 1: Re-sampling the data sets - Regression	53
3.3.3	Hypothesis 2: Adding cost-benefits	57
3.3.4	Conclusions	62
3.4	Comparison of Classification and Regression modelling approaches	64
3.4.1	Wilcoxon test: Best per metric and per data set	64

3.4.2	Post-hoc Nemenyi test: Average Ranks	66
3.5	Conclusions	69
4	Optimal Trading Signals	71
4.1	Introduction	71
4.1.1	Relationship with Activity Monitoring	72
4.2	Definitions and Concepts	73
4.3	Optimal Positions	74
4.3.1	Long Positions	74
4.3.2	Short Positions	78
4.4	Positive Activity for Trading	80
4.5	Optimal Signals as a Trading Benchmark	84
4.5.1	How to use the Optimal Benchmark	85
4.5.2	Standard Metrics for Evaluating Trading Systems	85
4.5.3	An Application to Evaluate Real Trading Signals	88
4.6	Conclusions	94
5	Conclusions and Future Work	97
5.1	Conclusions	97
5.2	Future Work	99
	Appendices	101
A	Actionable Forecasting - Generic Tasks	103
B	Actionable Forecasting - Trading Tasks	107

List of Figures

1.1	PLR used in Luo and Chen (2013) to obtain some “optimal” turning points	6
2.1	Relative frequencies on a standard random forest classification model with and without the cost-benefits	18
2.2	Best classification variant with and without costs per task: Accuracy	19
2.3	Best classification variant with and without costs per task: Precision	21
2.4	Best classification variant with and without costs per task: Recall	21
2.5	Best classification variant with and without costs per task: F-Measure	22
2.6	Best classification variant with and without costs per task: Utility	23
2.7	Best classification variant with and without costs per model type	25
2.8	Top average ranking classification variants with and without costs: 3 classes	29
2.9	Top average ranking classification variants with and without costs: 10 classes	32
2.10	Best classification and regression variant per task: Accuracy	34
2.11	Best classification and regression variant per task: Precision	36
2.12	Best classification and regression variant per task: Recall	36
2.13	Best classification and regression variant per task: F-Measure	37
2.14	Best classification and regression variant per task: Utility	38
2.15	Top average ranking classification and regression variants: 2 classes	39
2.16	Top average ranking classification and regression variants: 3 classes	40
2.17	Top average ranking classification and regression variants: 5 classes	41
2.18	Top average ranking classification and regression variants: 10 classes	42
3.1	Best classification variant with and without SMOTE per task: Precision and Recall	50
3.2	Best classification variant with and without SMOTE per task: Return and Sharpe-Ratio	51
3.3	Best classification variant with and without SMOTE per model type	52
3.4	Top average ranking classification variants with and without SMOTE	54
3.5	Best regression variant with and without SMOTE per task: Precision and Recall	55
3.6	Best regression variant with and without SMOTE per task: Return and Sharpe-Ratio	56
3.7	Best regression variant with and without SMOTE per model type	57
3.8	Top average ranking regression variants with and without SMOTE per model type	58
3.9	Best classification variant with and without costs per task: Precision and Recall	60

3.10	Best classification variant with and without costs per task: Return and Sharpe-Ratio	61
3.11	Best classification variant with and without costs per model type	62
3.12	Top average ranking classification variants with and without costs	63
3.13	Best classification and regression variant per task: Precision and Recall . . .	65
3.14	Best classification and regression variant per task: Return and Sharpe-Ratio	66
3.15	Top average ranking classification and regression variants: Precision and Recall	67
3.16	Top average ranking classification and regression variants: Return and Sharpe-Ratio	68
4.1	A long position opened by a false alarm	74
4.2	Optimal long positions of S&P 500 between July of 2012 and April of 2013 .	77
4.3	Optimal short positions of S&P 500 between July of 2012 and April of 2013	79
4.4	Positive activity of S&P 500 between July of 2012 and April of 2013	82
4.5	Real Trading Signals for one year period on the S&P500 index	88
4.6	Optimal Trading Signals for one year period on the SP500 index.	91
A.1	“Median” classification variant with and without costs per model type . . .	103
A.2	Top average ranking classification variants with and without costs: 2 classes	104
A.3	Top average ranking classification variants with and without costs: 5 classes	105
B.1	“Median” classification variant with and without SMOTE per model type . .	107
B.2	“Median” regression variant with and without SMOTE per model type	108
B.3	“Median” classification variant with and without costs per model type . . .	109

List of Tables

2.1	Cost-Benefit matrix for a 5 class task.	10
2.2	Brief description of each task.	11
2.3	Description of the used data sets	12
2.4	Utility matrix for a 5 class task	14
2.5	Regression models used for the experimental comparisons	15
2.6	Classification models used for the experimental comparisons	15
3.1	Sample of pre-processed Apple stocks	47
3.2	Example cost-benefit matrix for Apple shares.	59
3.3	Illustration of the use of a cost-benefit matrix on forecasts	59
4.1	List of trading policies.	90
4.2	Scores obtained by the trading system and by the optimal benchmark during the year of 1990 of the SP500 index	92
4.3	Scores obtained by a second trading system and by the optimal benchmark during the year of 1990 of the SP500 index	93

Chapter 1

Introduction

Decision support systems have been receiving increasingly more attention and are currently widely used in many domains. A large variety of methods have been developed and used. One of the main focus of this thesis consists in a particular subset of problems in which these decision systems may be applied. Namely, when a decision/action needs to be taken based on the forecast of a certain numeric variable. We call this type of data mining tasks as “actionable forecasting” problems. This particular type of tasks is very common in many areas (medical, financial, biological, etc.) and there are two distinct modelling approaches that may be used to model these problems: (i) a regression model may be used to model and forecast the numeric variable, and in a second step a deterministic function may be used to map this prediction into a decision/action; (ii) instead of forecasting the numeric variable we may obtain (classification) models that directly predict the action/decision associated with each numeric value using the same predictors.

To the best of our knowledge, there is no study that performs any comparative analysis between the eventual advantages or disadvantages of both approaches. One of the main contributions of this thesis is properly describe and exhaustively compare these modelling approaches.

We have split our comparative study of these two alternative approaches to actionable forecasting in two parts: (i) the first involves generic tasks; and (ii) the second involves a particular instance of actionable forecasting problems - financial trading tasks. On both parts we have considered a very large set of modelling tools and parameter variants of these tools to make sure our conclusions are robust across these different forecasting methods.

In the first set of generic tasks up to nearly 200 modelling variants and 32 modelling tasks were considered, with the number of decisions/actions per data set being $\{2, 3, 5, 10\}$.

In the second study we addressed a very specific application domain that fits the structure of actionable forecasting problems - financial trading. Given the relevance of this domain we have dedicated a full chapter just to analyse both modelling approaches on this application. Automatic trading systems are responsible for a large percentage of nowadays trades in financial markets. Risky decisions must be made with potentially high profits

or losses as a consequence. Any new study/discovery regarding trading systems may be crucial for any investor, motivating our analysis for this particular application. Therefore, the second main contribution of the thesis is to conduct another exhaustive experimental study to evaluate the advantages and disadvantages of both modelling approaches in the context of financial trading.

Our last contribution in this thesis is also related with financial trading. Namely, we address the key issue of how to evaluate financial trading decisions. Given a historic record of trading signals produced by some trading system, it is not an easy task to evaluate the performance of that trading system. The most common approaches is to have an investor analysing several metrics at the same time and deciding whether the observed scores are good enough for his own preference criteria. This procedure is very subjective and most of the trading metrics can not be adapted to the investor's trading policy (level of risk aversion as well as the target return per trade). In the thesis we propose a benchmark driven by the preference criteria of the investors against which we can compare any concrete trading system. This benchmark is derived based on a proposed formalisation of financial trading as a special instance of some data mining tasks known as activity monitoring. This formalisation and the derived benchmark provide new tools for looking at the performance of trading systems that we claim to be more interpretable for investors as they directly incorporate the investors' preference bias in terms of trading results.

1.1 Literature Revision

In this section we describe some of the work that has been carried out within the scientific areas that the thesis is related with. We divide the literature revision in two parts, one for the actionable forecasting tasks and a second one regarding the evaluation of a trading system.

1.1.1 Actionable Forecasting

To the best of our knowledge there are no research works directly comparing the two modelling approaches we are proposing for actionable forecasting tasks. There is some research regarding the usage of a single or a set of models using the same approach in a specific task, but never considering models of both approaches altogether.

Nevertheless, there are some concepts that are strongly related with the problem of Actionable Forecasting and thus we provide here a short review of some of the main works in these areas. In Actionable Forecasting tasks the ultimate goal is to predict an action/decision. However, there may be classes/decisions more important than others, or there may exist information on some costs and benefits associated with each decision. Another related problem is that of unbalanced distributions of the target variable in the context of prediction models. Frequently, in decision making, some of the decisions are less common, and moreover these tend to be more relevant. [Branco et al. \(2015\)](#) presents

a survey of existing techniques for handling these situations of unbalanced target variables. Although most of the existing work considers classification tasks (nominal target variables), this paper also describes methods designed to handle similar problems within regression tasks (numeric target variables). Regarding the problem of rare classes in the target variable [Weiss \(2004\)](#) and [Weiss \(2005\)](#) are good references on this topic.

We were not able to find generic studies on the problem of actionable forecasting. Still, there are a few works that address related problems on specific domains, such as medical, trading or other types of tasks. Nevertheless this type of research does not address the specific question we are aiming in this thesis: what is the best general methodology to address this type of decision making tasks.

Regarding the specific area of financial trading, since we devote a whole chapter to this type of tasks, we will now list some work that is somewhat related with our work. In terms of the usage 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, it requires a large effort in terms of model tuning and can be computationally demanding. [Gençay \(1999\)](#) 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 et al. \(2011\)](#); [Shin and Ghosh \(1995\)](#) and [Sermpinis et al. \(2012\)](#) have proposed variants and generalisations 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 model in this area are the k-nearest neighbours (KNN). [Gençay \(1999\)](#) 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 et al. \(2012\)](#) has 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 an analogy to the trading problem, i.e. detecting a buy or sell signal the earliest possible to obtain the maximum possible profit. In this article, this model achieved very optimistic and positive results. Support Vector Machines (SVM) and Multivariate Adaptive Regression Splines (MARS) have been the subject of study by [Kao et al. \(2013\)](#). 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.

[Fernandes \(2002\)](#) has compared the usage of several regression models in financial time-series forecasting. This author has tested regression trees, linear models, neural networks,

etc., in several financial data sets, and has also considered the bootstrap aggregation of those models. Evidence was found for arguing that these regression models can outperform the baseline benchmarks such as the random walk model or the average model, with statistical significance.

In terms of using classification models in the context of financial trading we have found fewer research works. [Chang et al. \(2009\)](#) 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 implemented cost matrices in 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 suggestion to buy some assets when we should sell it, 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 \(2009\)](#) contains a very detailed state of art regarding stock market forecasting techniques. The authors list the work carried out by several researchers. Namely, for each referred work the authors list the used data sets, the chosen input variables and, most importantly, a summary of all the used modelling 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 none of the referred works we have seen a comparison between regression models and classification ones in the context of financial trading. Several techniques are explored but the answer to the question proposed in the thesis seems to be absent from the research works we have encountered in our review of the state of the art in these areas. Moreover, it seems regression models are being more frequently used than the classification ones to address trading tasks.

1.1.2 Evaluation of Trading Performance

Another major topic of this thesis is the problem of evaluating a trading system. Whether a trading system tries to forecast the future variation of the prices or directly the final trading decision, the ultimate goal of these systems is to make the correct (and more profitable) trading decisions. In this context, from an investor perspective it is not very interesting to evaluate a system through metrics like the Root Mean Squared Error (in case there was a regression model trying to forecast the future price) or the Accuracy (how many correct decisions did the trading system made), but rather what were the

financial results of the decisions taken by the system. This is typically checked through the analyses of several specific trading metrics over a testing period. By doing so, the investor may for instance check the return and the risk associated with the trading system, which are measures that are absolutely crucial for investors. [Pardo \(2011\)](#) is a well known reference that includes some performance summaries for trading systems, using trading metrics such as the total net profit, the ratio between the average profit over the average loss, maximum draw down, profit factor and several others. The Sharpe ratio is a well known metric used to describe the risk associated with a trading system. There are more helpful measures to evaluate the risk associated with a trading system, but [Ferruz et al. \(2006\)](#) has seen that there is a strong correlation (higher than 0.9) between all possible combinations between the Sharpe, Treynor and Jensen risk measures when dealing with Spanish investment funds. This suggests that even though we may use several different risk measures, we will eventually obtain similar conclusions among them. Therefore, we will give special focus to the Sharpe ratio during the thesis. Unarguably, it has been one of the most used performance criteria to evaluate traders. It has been thoroughly studied by several researchers, namely by [Bailey and Lopez de Prado \(2012\)](#), [Lo \(2002\)](#) and [Pav \(2014\)](#). Essentially, this metric is able to check whether the returns of a portfolio are due to smart investment decisions or a result of excess risk. Its formula depends on a benchmark, where usually the null benchmark is chosen¹.

Despite the existence of several measures either characterising the profit or the risk of a trading system, only when considering a large set of metrics together one can properly infer about the quality of the system. [Folger \(2012\)](#) presents a possible guide on how to evaluate the performance of a trading system, where several metrics are considered. The author strongly advises to go over a thorough study regarding a vast number of metrics, involving the profit and the risk of the trading system.

This established procedure to evaluate the performance of a trading system, and the fact that most trading metrics have no dependency on the investor preference criteria in terms of profitability and risk, motivates the following question: if an investor A is willing to accept higher risk for higher returns and, on the other hand, another investor B prefers a more conservative policy, how can they determine if the performance of a certain trading system is the best for their preferences just by looking at these standard metrics? Answering this question is not easy as the metrics do not tell the investor how far are they from the optimal trading record from the perspective of their criteria. [Luo and Chen \(2013\)](#) has scratched the surface of this question, by using the piecewise linear representation (PLR) to obtain, on fully known data, what apparently would be the near-perfect points to trade. Even though it was not mentioned in his work, this could be used as some optimal benchmark to evaluate a trading system. However, this method depends on a very complex and hard to interpret parameter that will increase the number of optimal

¹Typically, the null benchmark is a trader that has no signals, i.e., holds the position all the time leading to zero profits and zero losses

turning points the lower the parameter gets. Figure 1.1 illustrates the usage of the PLR to obtain the “optimal” benchmark for a specific time-series and for different values of δ .

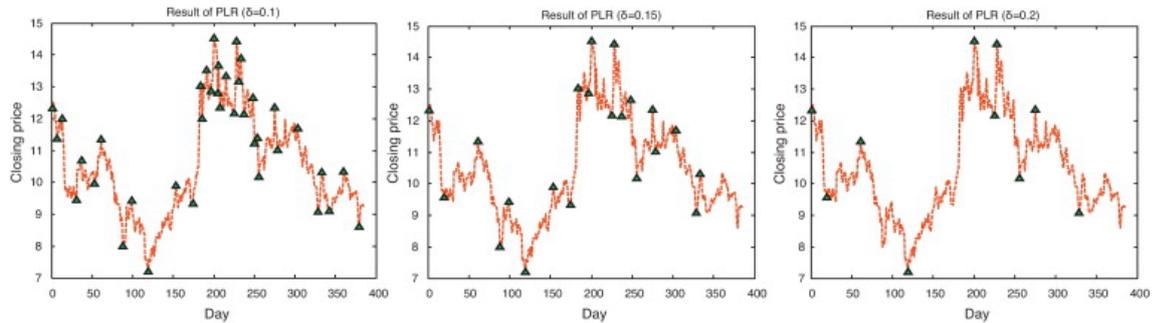


Figure 1.1: PLR used in [Luo and Chen \(2013\)](#) to obtain some “optimal” turning points

With the exception of this work that is slightly related with this concept of optimal benchmark, we have not found any other research literature that addresses this important question from the point of view of investors. This issue is one of the main contributions of the thesis.

1.2 Thesis Organisation

This thesis is organised as follows. Chapter 2 presents the concept of *Actionable Forecasting*, one of the main contributions of the thesis. We describe this type of applications and propose two alternative methods to address these problems. We study their advantages and disadvantages and present the results of an extensive empirical comparison of these alternatives using a large set of generic tasks and an extensive set of modelling tools.

In Chapter 3 we focus on one particular instance of the actionable forecasting tasks - financial trading. This is an application with high impact in our society. The characteristics of this application requires some modifications to our proposals. We compare the two alternative methods to solve actionable forecasting tasks in this particular application, with the goal of checking if their particularities change somehow the conclusions we have reached for the general case.

Chapter 4 describes our final contribution that addresses the evaluation of trading systems. We propose a formalisation of financial trading as a special instance of activity monitoring data mining tasks. Based on this formalisation, we provide means of obtaining the optimal trading actions given the economic preferences of an investor. We describe this optimal benchmark and apply it to evaluate the trading record of a real investor, highlighting the advantages of our proposal.

Finally, in Chapter 5 we briefly describe the main conclusions and contributions of this thesis, and outline possible directions for future research developments related with our thesis.

Chapter 2

Actionable Forecasting

Actionable Forecasting can be defined as the process of predicting the right action, when the action itself is a function of a predicted numerical variable. This problem is very common in several distinct areas, such as medicine, trading, industry market, etc. For instance, consider the process of deciding which type of treatment should be applied to a patient with cancer based on the future volume of the tumour. Naturally, there is no way of knowing exactly this value, however a decision must be made. The “simplest” procedure would be to somehow predict the growth of the volume and based on that forecast make a decision. As we will see, there are other approaches to deal with this problem and one of the main contributions in the thesis is to properly compare two of the most used approaches. Another example application, whose importance is so high that will be separately studied in a further section, is financial trading. As an investor, deciding whether to open a short/long position or to hold takes serious risks. In principle, if the investor could be sure if a certain asset would increase or drop its value, then the decision would be straightforward to make. However, a decision must be made without knowing the numerical value of the asset in the future. Once again, the “simplest” procedure of somehow predicting the numerical value and then make the decision would work, but other approaches are also valid.

2.1 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 \mathbf{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,

$$f: \mathbb{R}^p \rightarrow \mathbb{R}$$

$$\mathbf{x} \mapsto f(\mathbf{x}).$$

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,

$$g: \mathbb{R} \rightarrow \mathcal{A} = \{a_1, a_2, a_3, \dots\}$$

$$Y \mapsto g(Y).$$

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 \mathbf{x} into Y , i.e. a data set that can be used to learn an approximation of the function f . We call this class of problems *actionable forecasting tasks*.

At this stage it is important to remark that other related problems exist. For instance, there may exist applications for which g is also uncertain, or it may even vary with time. These alternative decision making scenarios are not addressed in this thesis. Our goal is to address the specific decision tasks that was described previously in the above formalization.

2.2 Proposed approaches

Given that the variable Y is numeric, a prediction of its value could be obtained using some existing multiple regression tool. This means that given a data set $D_r = \{\langle \mathbf{x}_i, Y_i \rangle_{i=1}^n\}$ we can use some regression tool to obtain a model $\hat{r}(\mathbf{x})$ that is an approximation of f . From an operational perspective this would mean that given a test case \mathbf{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, i.e. $\mathbf{q} \mapsto \hat{r}(\mathbf{q}) \mapsto g(\hat{r}(\mathbf{q}))$.

Given the deterministic nature of g , we can use an alternative process for reaching decisions. More specifically, we can build an alternative data set $D_c = \{\langle \mathbf{x}_i, g(Y_i) \rangle_{i=1}^n\}$, 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, i.e. 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

\mathbf{q} to directly estimate the correct decision by applying the learned model to the case, i.e. $\mathbf{q} \mapsto \hat{c}(\mathbf{q})$.

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, while 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 chapter is to compare these two approaches in the context of this type of decision tasks.

2.2.1 The Classification Approach: A limitation

A problem that may hinder the performance of the classification modelling approach is presented and discussed. These models face a theoretical problem as these algorithms do not distinguish among the different types of errors. Consider a task with three possible classes/decisions C_1, C_2, C_3 , that result from a deterministic mapping of a numerical response Y , where C_1 corresponds to a value lower than k_1 , the class C_3 to a value higher than $k_2 > k_1$ and the class C_2 to the the values in between. Typically, confusing C_1 with C_3 is a more serious mistake than C_1 with C_2 . However, the way a classification model is built makes all the errors equally weighed, as they assume a nominal target variable with no ordering among its values.

This problem led us to consider an alternative to our base modelling approaches . We have considered a frequently used approach to this issue. Namely, we have used a cost-benefit matrix that allows us to distinguish between the different types of classification errors (e.g. [Torgo and Gama \(1997\)](#)). Using this matrix, and given a probabilistic classifier, we can predict for each test case the class that maximises 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 generic tasks. Correctly predicted decisions are rewarded with 1. On the other hand, any wrong predictions are penalised as minus the number of classes away from the correct one, e.g. forecasting as C_5 a C_1 class is penalized with -4 , while predicting C_2 on a true C_3 will receive a score of -1 . Table 2.1 shows an example of such a cost-benefit matrix for a generic data set with 5 possible actions

	C1	C2	C3	C4	C5
C1	1.00	-1.00	-2.00	-3.00	-4.00
C2	-1.00	1.00	-1.00	-2.00	-3.00
C3	-2.00	-1.00	1.00	-1.00	-2.00
C4	-3.00	-2.00	-1.00	1.00	-1.00
C5	-4.00	-3.00	-2.00	-1.00	1.00

Table 2.1: Cost-Benefit matrix for a 5 class task.

We have also thoroughly tested the hypothesis that using cost-benefit matrices to implement utility maximisation would improve the performance of the models. The results will be presented after the experimental methodology is described.

2.3 Material and Methods

Having described the concept of the Regression and Classification approaches for the Actionable Forecasting problem, our next goal is to compare both of them in different experimental settings. We have chosen an empirical comparison because it may be difficult, perhaps impossible, to compare them in a theoretical way since each task and each modelling tool present their own theoretical properties, with some being extremely complex and distinct. Hence, any chance of incorporating several distinct tasks and models on the same study would not be viable.

The main issues involved in the experimental comparison will now be presented. Firstly, the tasks that will be used in our study are detailed, describing both the available data sets and the variables involved in the process. Secondly, we address the important decision of how the alternatives will be evaluated and compared by describing the evaluation metrics used in our experiments. Finally, the modelling tools that will compose each modelling approach will be listed.

All the work of the thesis was performed using the R programming language [R Core Team \(2014\)](#).

2.3.1 The Tasks

In this chapter the goal is to perform a generic study, minimising the eventual influence of problems such as the class imbalance or the existence of decisions much more important than others. These questions will be addressed in the next chapter, namely when dealing with trading tasks.

In the experiments of this chapter several regression tasks will be considered in which the g function, that maps the numerical response into actions, will consist of defining thresholds that will make all the respective classes (i.e. decisions) somewhat balanced in terms of frequency. Eight data sets of completely different contexts were used, whose number of observations vary between 1503 and 10617 (some brief details about each data

set are given in Table 2.2). Each data set will lead to four different tasks, resulting from the application of four different variants of the g function. Essentially, for each data set, a version with 2, 3, 5 and 10 classes/decisions is formed. The point is to ensure a rich study by not only having distinct contexts, but also a diversified set of possible actions per context. Behaviours such as one approach being preferable to predict a smaller set of actions while the other approach being fitter to model data sets with more actions may thus be captured.

Data set (Nr.Cases; Nr.Predictors)	Description
Abalone (9;4177)	Predicting the age of abalone from physical measurements. The age of abalone is determined by cutting the shell through the cone, staining it, and counting the number of rings through a microscope.
Airfoil (6;1503)	The NASA data set comprises different size NACA 0012 airfoils at various wind tunnel speeds and angles of attack. The span of the airfoil and the observer position were the same in all of the experiments. The goal is to predict the pressure.
Concrete (9;1030)	Predict the actual concrete compressive strength for a given mixture under a specific age that was determined from laboratory.
Bank8FM (9;4499)	A simulation of how bank-customers choose their banks. Tasks are based on predicting the fraction of bank customers who leave the bank because of full queues.
CCP (5;9568)	The data set contains data points from a Combined Cycle Power Plant. Features consist of hourly average ambient variables Temperature, Ambient Pressure, Relative Humidity and Exhaust Vacuum to predict the net hourly electrical energy output of the plant.
CpuSmall (13;8192)	Relative CPU Performance Data, described in terms of its cycle time, memory size, etc.
Delta Ailerons (6;7129)	The task of controlling the ailerons of a F16 aircraft
Naval (13,8000)	A numerical simulator of a naval vessel produced a 16-feature data set containing the Gas Turbine measures at steady state of the physical asset. The goal is to predict Gas Turbine Turbine decay state coefficient

Table 2.2: Brief description of each task.

Table 2.3 gives some information of the created data sets. All consisted of tasks whose response variable was originally numeric. However, a new classification variable was added by mapping the numerical response using the threshold function g . In this way, regression models can be used to predict the numerical response, where the prediction will be applied in the known function g to construct a decision, while classification models can be directly used to predict the “artificial” variable. Note that the relative frequency of each class is not the same across all the data sets with the same number of classes.

	Nr Actions	Min Freq %	Max Freq %
Concrete10	10	9.30	11.50
Concrete5	5	16.40	24.60
Concrete3	3	31.30	36.70
Concrete2	2	40.10	59.90
Naval10	10	7.40	11.80
Naval5	5	18.90	23.30
Naval3	3	30.50	38.80
Naval2	2	42.10	57.90
Airfoil10	10	10.00	10.00
Airfoil5	5	20.00	20.00
Airfoil3	3	30.80	36.90
Airfoil2	2	36.50	63.50
CCP10	10	10.00	10.00
CCP5	5	20.00	20.00
CCP3	3	25.60	39.90
CCP2	2	38.80	61.20
CpuSmall10	10	8.60	12.40
CpuSmall5	5	18.80	21.00
CpuSmall3	3	27.40	36.90
CpuSmall2	2	48.70	51.30
Bank8FM10	10	10.00	10.00
Bank8FM5	5	20.00	20.00
Bank8FM3	3	32.30	34.70
Bank8FM2	2	49.40	50.60
DeltaAilerons10	10	2.30	21.90
DeltaAilerons5	5	14.50	26.70
DeltaAilerons3	3	21.90	46.90
DeltaAilerons2	2	46.90	53.10
Abalone10	10	4.50	16.50
Abalone5	5	10.70	26.80
Abalone3	3	20.10	45.30
Abalone2	2	49.80	50.20

Table 2.3: Description of the used data sets. The last two columns correspond respectively to the relative frequency of the least and most frequent classes/actions.

2.3.2 Evaluation Metrics

Knowing the tasks that will be used in the experimental study, we encounter the need to define how the methods of each proposal should be evaluated in the upcoming experimental study. The simplest, yet intuitive way that is widely used to evaluate models that are producing decisions is by looking at the ratio of correct predictions over the total number of predictions. Clearly, it gives some strong information regarding the quality of the models. However, in the actionable forecasting problem, one may give different levels of importance to each possible decision or unequal costs may be assigned to every possible pair of predicted/corrected decisions (see the work of Branco et al. (2015) that addresses the importance of using cost-benefit matrices, as well as the articles of Weiss (2004) and Weiss (2005)). Therefore, different metrics should be considered. Standard classification metrics will be used, whilst very specific and contextual metrics, namely the return and risk metrics for financial problems, will be addressed in the next chapter.

The following metrics will be used in our experimental comparisons of this chapter.

Note that the Macro-average versions of certain metrics will be used because we do not want to analyse each class separately but rather by its overall performance across the domain of the response variable. The following metrics are described with more detail, including the macro version of certain metrics, in the work of [Sokolova and Lapalme \(2009\)](#).

- **Accuracy:** It consists of the percentage of correct predictions over wrong ones;
- **Macro Recall:** Generically, it measures the ratio of how many cases of a certain class were properly captured by the model. More specifically, the recall of a certain class is given by True Positives / (True Positives + False Negatives). The macro Recall consists of averaging the Recall of all classes;
- **Macro Precision:** On the other hand, this metric measures the correctness of the model when it predicts a certain class. High values of prediction for a class A indicates that every time the model predicts A , it should be most likely a correct prediction (with a chance equal to the Precision value for this class). More specifically, the precision of a certain class is given by True Positives / (True Positives + False Positives). The macro Precision consists of averaging the Precision of all classes. Models with a high value of precision are considered to be safer, since most classes will only be predicted when the model is “confident” on those predictions;
- **Macro F-Measure:** The F-measure is a trade-off between the recall and precision of a class, whose weight is determined by a coefficient β . Namely, when $\beta = 1$ both metrics are equally weighted. This metric is particularly useful to ensure that model was not overly optimized for one of the criteria by severely compromising the other one. The formula is given by $(1 + \beta^2) (\text{precision} \times \text{recall}) / (\beta^2 \times \text{precision} + \text{recall})$. The macro F-Measure uses the average recall and precision in the previous formula;
- **Utility Score:** In a generic sense, this metric allows the user to give a reward or penalty for every possible pair of true value/forecast. This way, certain classes may be considered more important than others or there may be some classes that we only want to forecast when the forecasting system is quite confident on that forecast. It may be also used to give more emphasis to under represented classes (higher rewards for predicting correctly rarer classes). In our particular application, we will chose an utility matrix such as we may evaluate not only the correctness of the model but also how far each decision was from the correct one. This notion of distance between categorical values makes sense since the decisions are obtained from the application of thresholds onto a numerical variable, thus leading to ordered classes. Therefore, we can consider the “distance” between the class C_1 and C_2 equal to 1, the “distance” between the class C_1 and C_3 equal to 2, etc. Essentially, for every prediction the model will be either rewarded or penalized according to the following matrix (illustrated for a 5 class example).

	C1	C2	C3	C4	C5
C1	1.00	-1.00	-2.00	-3.00	-4.00
C2	-1.00	1.00	-1.00	-2.00	-3.00
C3	-2.00	-1.00	1.00	-1.00	-2.00
C4	-3.00	-2.00	-1.00	1.00	-1.00
C5	-4.00	-3.00	-2.00	-1.00	1.00

Table 2.4: Utility matrix for a 5 class task. It is equal to the Cost-Benefit matrix presented on 2.1

Hence, every model can be given a score using this methodology, and the Utility Score will consist on the subtraction of the score obtained by the model under evaluation against a benchmark model that will predict everything as the most frequent class.

With this set of metrics, there are sufficient tools to compare the models by means of safety, by means of properly detecting the classes, by its overall predictive power and by their utility value. This should be enough to conduct a more detailed comparison between the proposed modelling approaches and analyse the strengths and weakness of either approach.

2.3.3 The Models

The tasks as well as the evaluation metrics have been presented. The last step before going into further detail about the experimental methodology is to describe and list the model variants that will fuel this analysis.

The choice of models was made with the goal of ensuring a certain diversity that precludes any model-dependent bias of the conclusions of our experiments. Several variants for each family of models (SVM, Random Forests, etc) were tested in order to make sure our conclusions were valid under different setups. Tables 2.5 and 2.6 show all the model variants used, specifically $76 + 97 = 173$ model variations were considered (classification and regression, respectively). Some of the following models are experimentally tested (on a large domain of tasks) in the work of [Ahmed et al. \(2010\)](#) . The total number of model variants is yet to be increased. Some variations of these standard models will be introduced later in order to try to solve some theoretical problems these models may have.

2.4 The Experimental Methodology

In this section we present the experimental methodology used in our comparative experiments of the two approaches to actionable forecasting. Given that the objective of this work is to study the advantages/disadvantages of either approach, both must be tested exactly under the same conditions.

To obtain reliable estimates of the metrics we have selected to compare the two approaches, we have used 10-fold Cross Validation. This methodology consists of splitting

Model	Variants	R Package
SVM	cost={1,5,10}, $\epsilon = \{0.1,0.05,0.01\}$, tolerance={0.001,0.005},kernel=linear	e1071 Meyer et al. (2014)
SVM	cost={1,10}, $\epsilon = \{0.1,0.05,0.01\}$, 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)

Table 2.5: Regression models used for the experimental comparisons. SVM stands for Support Vector Machines, KNN for K-nearest neighbours, NNET for Neural Networks and MARS for Multivariate Adaptive Regression Spline models

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}, $\epsilon = \{0.1,0.05\}$, 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 et al. (2013)

Table 2.6: Classification models used for the experimental comparisons. SVM stands for Support Vector Machines, KNN for K-nearest neighbours, NNET for Neural Networks and MARS for Multivariate Adaptive Regression Spline models

the data set into 10 equally sized parts, whereas every combination of 9 of those parts will be used to train the model allowing the remaining part to be used as a test set. This will allow every model variant to be evaluated ten times, where every observation falls one and one time only into some test set. Finally, those ten scores will be used to approximate the true distribution of the score for the tested model thus enabling the use of statistical tests to compare different models for the same metric. Note that a distribution is inferred for every model, every data set and every metric. The Cross Validation methodology is exhaustively analysed in the work of Arlot et al. (2010)

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 test the significance of the differences (in oppose to the t-test that assumes normality that may not be true overall; For the interested reader, see the work of [McCrum-Gardner \(2008\)](#) which details in which conditions should each statistical test be used). 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 differences between the average ranks of the k models across the t tasks.

The Wilcoxon signed rank test is a non-parametric test whose null hypothesis states that the median difference of both distributions is zero. It can be used to compare two matched samples with the same length. Conceptually, it considers the absolute difference of paired observations and ranks them (the minimum difference has rank 1, etc) while on a second step, those ranks will be multiplied by the signal of the differences. Later, the sum of signed ranks is considered, denoted by \mathcal{W} , whose value can be easily interpreted. Intuitively, if \mathcal{W} is a high absolute value, then the median of one distribution is considerably higher than the other (in an extreme case, all the signed ranks have the same signal, implying the values of one sample to be overall smaller than the other, leading to a high absolute value of \mathcal{W}). On the other hand, if this value is close to zero, then the median difference should be quite low. Theoretically, W tends to follow a known distribution and if W falls within the 95% more common values then we cannot reject the null hypothesis that states unequal median values, whilst if it falls in the remaining 5%, we can reject the null hypothesis.

The Friedman test followed by a post-hoc Nemenyi is a slightly more complex statistical test. Conceptually, the Friedman test ranks the models for each data set separately, the best performing algorithm getting the rank of 1, the second best rank 2, etc. The null hypothesis states that the average ranking of all algorithms should be equal. The Friedman statistic follows a certain distribution which enables the construction of a p-value for this test. If the null hypothesis is rejected, then we move on to the Nemenyi's test. One should take in consideration that this initial test checks if the average ranking of all the model variants are equal. This test is important, since if the null hypothesis cannot be rejected, then we can save up the work of testing if the average rankings of every pair of models statistically different according to a Nemenyi's test. This new test, will conduct a similar study, but by directly comparing one model against the other. The structure is quite similar to the Friedman's test, but the embedded distributions for the statistics have different parameters.

A very important remark regarding the Friedman test followed by a post-hoc Nemenyi related to the safety of the test. This test will only give significant results for algorithms whose performance is inarguably different among them. For instance, across 10 tasks and 12 models, if one model obtains the first rank for all the tasks and another models obtains all the time the third rank, this test would fail to consider both models as significantly different. Therefore, even if this test ceases to reject its null hypothesis, we should not

immediately consider that all the algorithms have the same performance.

The R programming language will be used to perform the experimental study, with special attention being paid to the package “performanceEstimation” [Torgo \(2013\)](#). Particularly, it allows several model variants to be run over different tasks and metrics, using experimental methodologies such as Cross Validation to produce reliable estimates of the performance of each model on each task. It also facilitates the process of comparing the performance of any set of models.

2.5 Analysis of the usage of cost-benefit matrices

We have seen before that approaching our target tasks using classification approaches has the potential problem of ignoring the implicit ordering among the labels that result from the discretization of the initial numeric response. We have put forward the hypothesis that this drawback could be overcome by using cost-benefit matrices. In this section we check the validity of this hypothesis by comparing different models with and without the use of cost-benefit matrices.

This study will be divided into three parts. The first part (Section [2.5.1](#)) of our experimental study compares the best modelling approach of each alternative (costs vs no-costs) per data set and checks which one is better, making use of an Wilcoxon test. The second part (Section [2.5.2](#)) presents a similar study but segmented per type of model, allowing more precise conclusions to be drawn. Finally, in the last part [2.5.3](#), the average rankings across all the tasks of the top five modelling tools of each modelling approach are considered and compared, making use of a Post-hoc Nemenyi test.

Our main expectations in terms of which benefits the usage of costs will bring to our models consist of decreasing the seriousness of the errors of the models. This means we are trying to “teach” them the implicit ordering among the values of the nominal target variable (the decisions). Therefore, the models should only try to predict an extreme class (C_1 or C_{10} for instance) when they are really confident about it as making an error in such situation will increase the cost of the predictions a lot. Thus, we should see more predictions on the central classes. It is not clear what will happen in terms of the accuracy of the models. Actually, the standard models are already optimised for this metric, meaning that using costs may eventually harm this score. On the other hand, in terms of the utility score we are expecting to see some boost.

Regarding the macro versions of Precision and Recall, it is not clear what to expect. We should expect more forecasts on the middle classes, suggesting an increase of the Recall level on those classes (since the chance of capturing these events increases) with an eventual decrease of Precision (since with more predictions on these classes, the chance of making a mistake also increases). The opposite behaviour should occur on the extreme classes, since with less forecasts there the Recall should decrease and the Precision increase. When averaging those effects (macro), it becomes hard to predict the expected final behaviour.

As a side note, the differences between the standard models and the ones with costs should become more evident with the increase of the number of possible actions/decisions. If there are more classes to be predicted, then there will be more chances to confuse distant classes, allowing the cost-benefit matrices to have a higher influence.

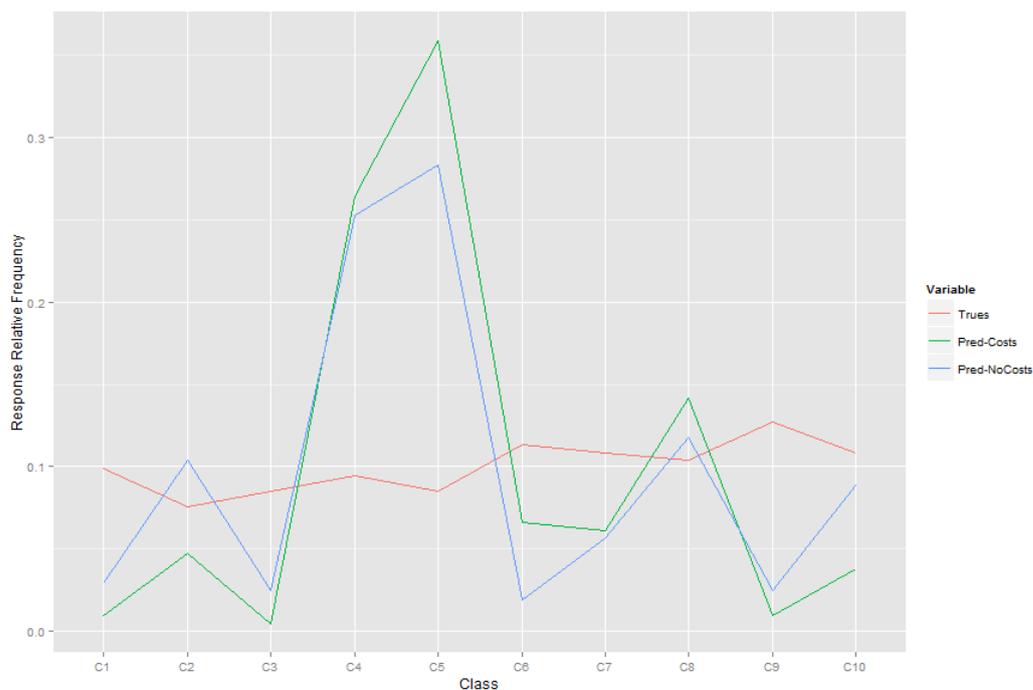


Figure 2.1: Analysis of the relative frequencies on a standard random forest classification model (default specifications of the used package) with and without the usage of cost-benefit matrices.

Before starting the main analysis, we illustrate the effect of the use of cost-benefit matrices with a simple example on a Actionable Forecasting tasks. We have considered a regression task and categorised the numeric response into 10 ordered classes. In Figure 2.1, we have the relative frequencies of the predictions of a RandomForest model (default specifications used), with and without the usage of cost-benefit matrices. Eighty percent of the data set was used as the training set, with the remaining part being used as the testing sample. The relative frequency of the model with costs is superior in the central classes and lower on the extreme ones. Even though it seems that the model without costs is more frequently closer to the true relative frequency, one should remember the embedded numeric variable on the response target, implying that we should not only look for the right decision but also try to minimise the severity of a mistake when it occurs.

2.5.1 Wilcoxon test: Best per metric and per data set

The first part of this study consists of performing a Wilcoxon test between the best classification variant without costs against the best one with costs per data set and per metric.

In other words, for each individual metric we are selecting the best model variant from the alternatives using cost-benefit matrices and comparing it against the best model variant not using these matrices. This study may provide some insight about the impact of the usage of this cost-sensitive approach on classification models, but one should keep in mind that we are only comparing the top performer of each approach (cost vs no-costs) and it may be dangerous to infer any global conclusion by merely looking at these top variants

In Figure 2.2, we can see the results of the comparison in terms of Accuracy. For each data set, we have used the Wilcoxon test to check the statistical significance of the observed differences. The presence of an asterisk implies that the null hypothesis of the statistical test was rejected, thus indicating that one model significantly outperformed the other. The red bars refer to the standard models while the blue ones are for the models using costs.

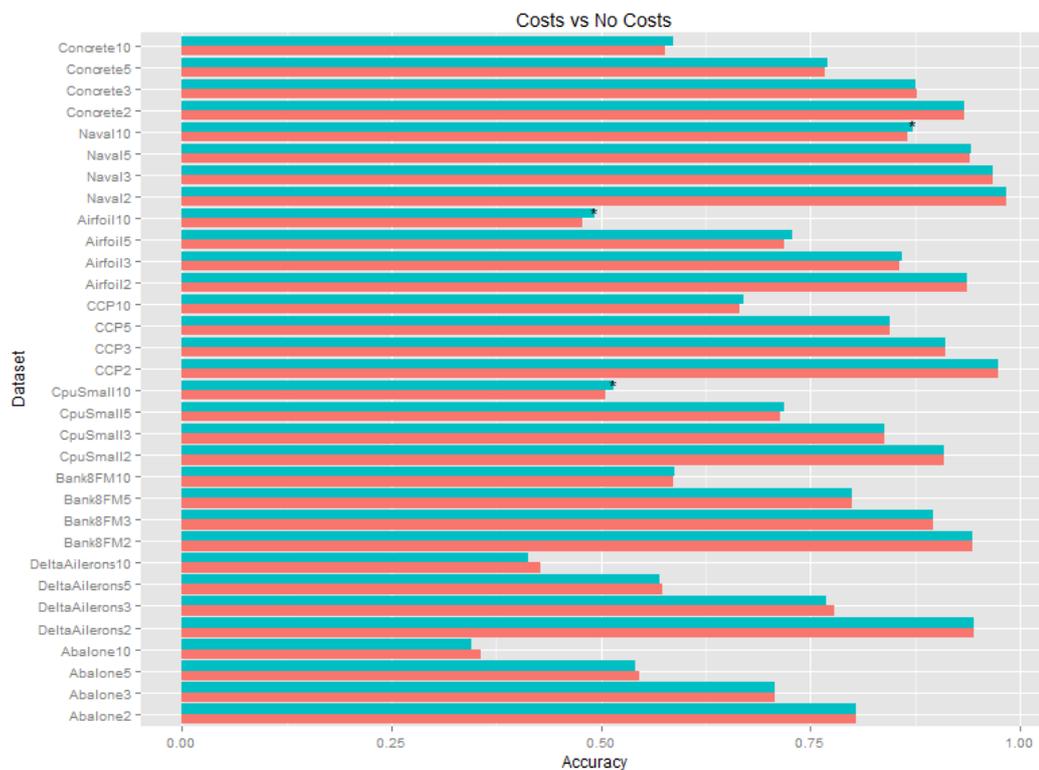


Figure 2.2: Best classification variant without costs (red) against the best classification variant with costs (blue) for the Accuracy score, where the presence of an asterisk implies that the respective variant performed significantly better than the one it is competing against, according to a Wilcoxon test with $\alpha = 0.05$.

There is a total of three significant wins of the model variants with costs against zero of the standard models. This comes as a surprise, since the standard models are optimised for this metric. However, only the best representative of each modelling approach is being considered, meaning that this behaviour may not occur across all the models

overall. Moreover, the three significant wins occur only in tasks with 10 classes. One possible explanation is that it is theoretically expected for the cost-benefit matrices to have a higher influence on tasks with more classes, where the implicit ordering among the class values is more important, providing more space for a positive impact for using costs.

In Figures 2.3 and 2.4 we see analogous plots for the Precision and Recall metrics. In terms of Precision, the model variants with costs have obtained several significant wins while some balance has been observed in the Recall metric. These graphs tell us that the new models (with costs) are increasing, in average, the precision level of each class individually without compromising their ability to properly detect, in average, more observations of each class when compared to the standard ones (without costs). These metrics may be quite useful in tasks whose classes of the target variable are not equally important.

This behaviour could be explained by the fact that the models with embedded costs will predict more cases as the middle classes¹. On the contrary, these models forecast an observation on the extreme classes (such as C_1 and C_{10}) only when there is a high level of confidence for that prediction. This happens because confusing those extreme classes would incur on a very large cost, thus leading to more “central” forecasts in order to reduce the expected cost.

Therefore, the precision on the “non central” classes should be quite high while the precision for the most central ones should decrease. On the other hand, due to more accurate, safer and consequently, fewer predictions on the extreme classes of the models with costs, the Recall will fall, with some increase being verified on the central classes. When considering the macro version of these metrics (the average across all the classes), our results suggest that using costs improves the Precision level without any serious change being verified on the Recall metric. This may be considered as a decent improvement when using cost-benefit matrices, though one should know the reasoning behind this improvement.

One should also note that most of the significant wins occur on tasks with 10 classes, the ones where the usage of costs may have a higher influence on the final predictions, thus increasing the differences between the models.

To sum up, the usage of costs is biasing the forecasts to the more central classes, allowing the predictions on the remaining classes to occur solely when there is a high degree of confidence. Therefore, we are seeing higher values of Precision when applying costs to the standard models. That could have come at the cost of some decrease on the levels of Recall, but that was not confirmed in our analysis. Even though we are just looking at the best model variant, we should see this behaviour, mainly the increase on the Precision metric, for most of the remaining models (something to be analyzed in the next parts of this study), since it is theoretically expected to see the models with costs

¹By middle class, we mean the classes that result from the mapping of the central values of the numeric intermediate variable. For instance, in a 10 class task, the middle ones would be the C_4 , C_5 and C_6 , since they would result from the most central values of the internal numeric target variable.

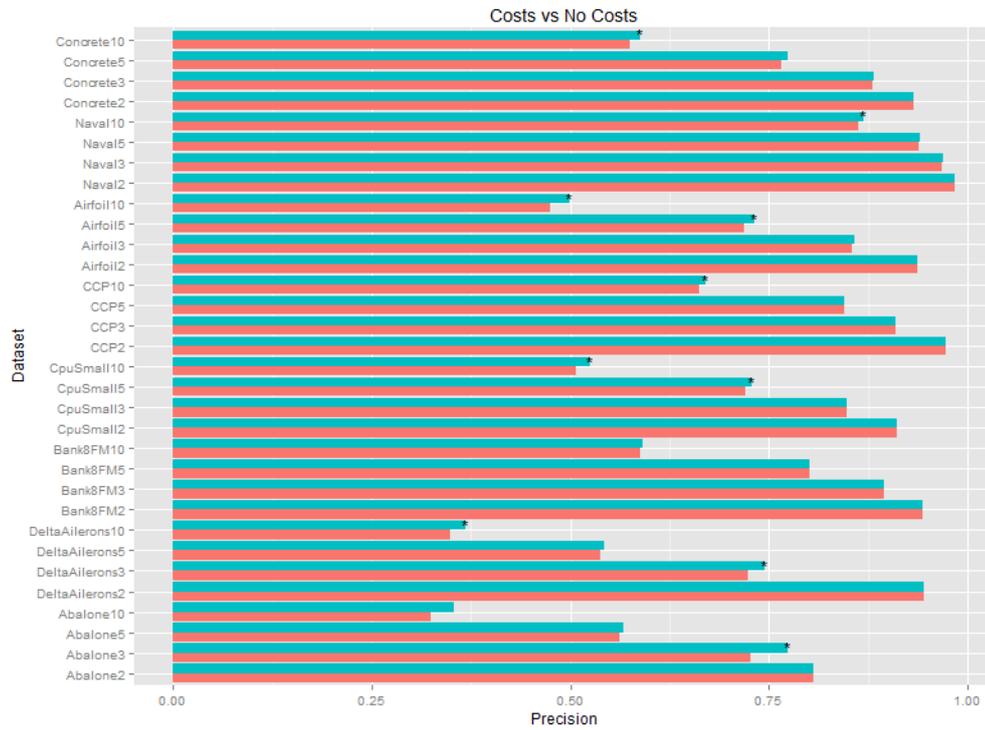


Figure 2.3: Best classification variant without costs (red) against the best classification variant with costs (blue) for the Precision score, where the presence of asterisk implies that the respective variant performed significantly better than the one it is competing against, according to a Wilcoxon test with $\alpha = 0.05$.

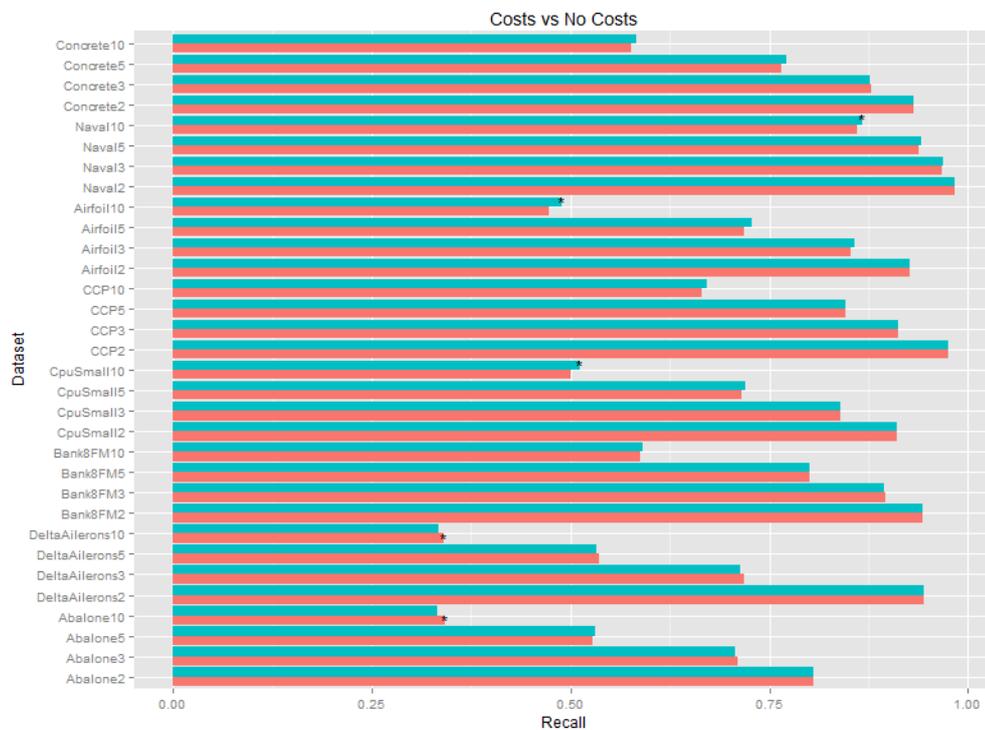


Figure 2.4: Best classification variant without costs (red) against the best classification variant with costs (blue) for the Recall score, where the presence of asterisk implies that the respective variant performed significantly better than the one it is competing against, according to a Wilcoxon test with $\alpha = 0.05$.

making decisions more often in the central classes in order to avoid the heavy costs given to confusing classes at maximum distances (such as C_1 and C_{10}).

Regarding the F-Measure, which is a metric that equally weighs Macro-Recall and Macro-Precision into a single metric, we should be seeing some trend favouring the models with costs, since they achieved better values of Precision without severely compromising the Recall. In Figure 2.5, we can see the results for the F-measure metric with $\alpha = 1$. Indeed, the results confirm our expectations, where once again, most of the significant wins occurred on the tasks with 10 classes.

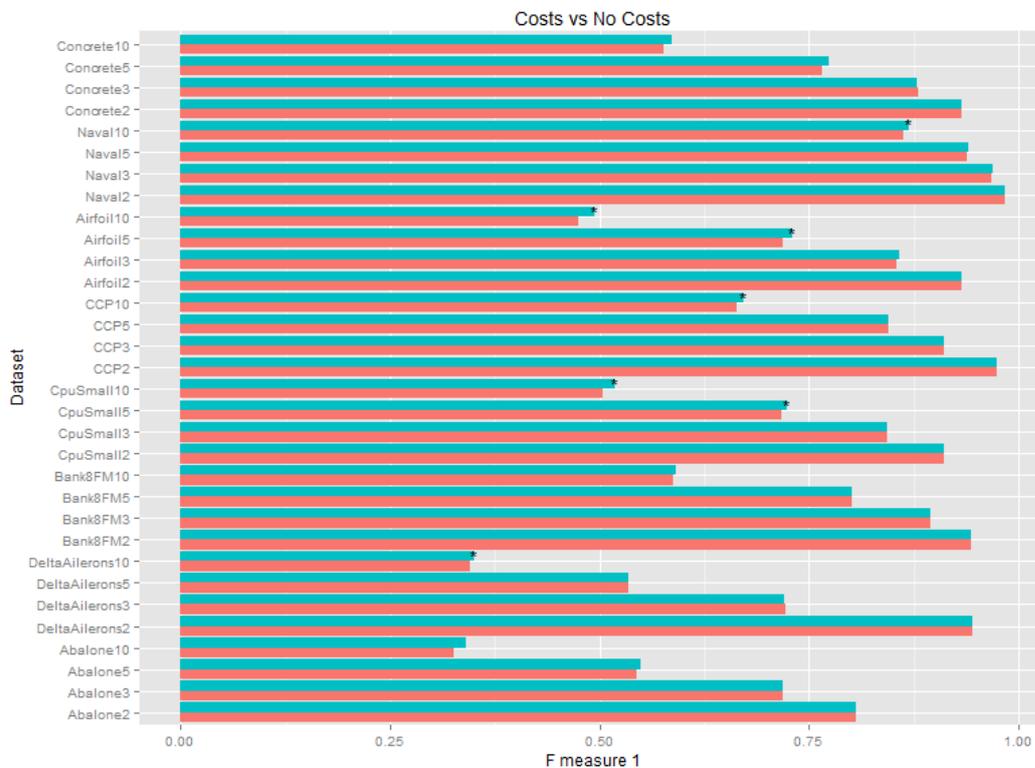


Figure 2.5: Best classification variant without costs (red) against the best classification variant with costs (blue) for the F-Measure score, where the presence of asterisk implies that the respective variant performed significantly better than the one it is competing against, according to a Wilcoxon test with $\alpha = 0.05$.

Finally, we analyse the Utility measure in Figure 2.6. There are 8 against 1 significant wins clearly demonstrating some trend favouring the models with costs. This should be expected, since the models using cost-benefit matrices are using information that should help in maximising the utility of their predictions. Hence, one may argue that it is not a fair comparison, but the point here is to empirically prove the expected advantage of the models with costs in terms of Utility, and that it may come without harming (or it could actually improve) some other metrics like the Accuracy and Precision. As in the previous cases, most of the significant wins using costs occur on the tasks with more classes.

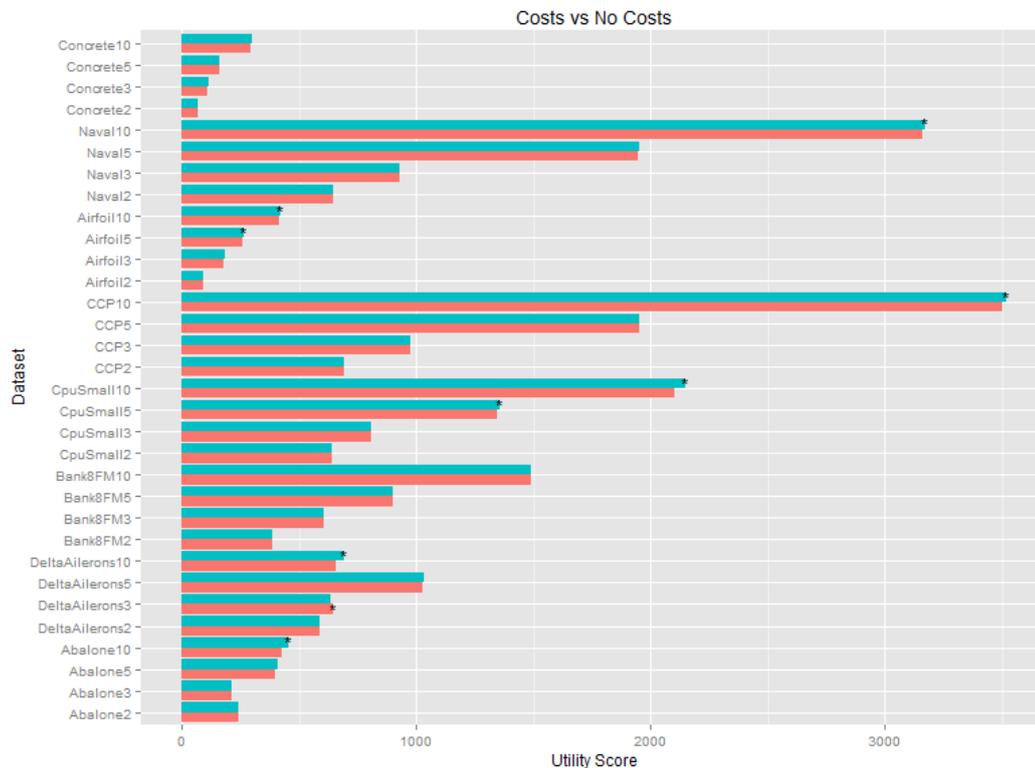


Figure 2.6: Best classification variant without costs (red) against the best classification variant with costs (blue) for the Utility score, where the presence of asterisk implies that the respective variant performed significantly better than the one it is competing against, according to a Wilcoxon test with $\alpha = 0.05$.

To sum up, in this first part of the analysis, we have seen that both approaches are somewhat balanced with some advantages for the usage of costs, at least when selecting the best representative of each modelling approach. In terms of Accuracy, there was some small advantage of the models using cost-benefit matrices whilst in terms of Precision the difference was more noticeable. Some equilibrium was observed in the Recall metric. Finally, when it comes to the Utility score, the new models are obtaining considerably better values, thus indicating that adding costs to the baseline models is worth considering and may add some value to the whole set of classification models.

In the next part of the study, we will move from comparing the best overall variants (with and without costs) to the best variants per modelling technique, such as the Support Vector Machines, Random Forests, etc.

2.5.2 Wilcoxon test: Best per metric, per data set and per type of model

In the second part of the analysis of the impact of the use of costs on the performance of classification models, we present a different type of study. Instead of selecting the best variant across all the models for the two alternatives (costs vs no-costs), we will select the

best for each type of model. This will allow us to check if the conclusions of the previous section still hold if we focus on a single algorithm (e.g. SVM or Random Forest).

Using the same type of graphs as in the previous section would entail a very large number of plots, so we have decided to use a different visual representation of the results of this experiment. Our solution is given in Figure 2.7. We divide the tasks according to the number of classes, meaning that the numbers 2, 3, 5, 10 at grey indicate which type of task is under analysis. The columns on the right indicate the metric being used to evaluate the models. On the left, we see the type of models, repeated for each metric. Each subplot illustrates the number of significant and non significant wins across the 8 tasks (there are 8 data sets with 2 classes, 8 data sets with 3 classes, etc) per type of model. Finally, the colours indicate the following:

- **Strong Blue:** Significant Wins for Models without costs;
- **Weak Blue** - Non Significant Wins for Models without costs;
- **Yellow** - Draws (meaning both top model variants had the same exact score);
- **Weak Green** - Non Significant Wins for Models with costs;
- **Strong Green** - Significant Wins for Models with costs.

Let us consider, for instance, the subplot for the Accuracy metric for data sets with 3 classes: i) We can see that the NNET (neural networks) type of model returns 1 significant win for the models without costs, 5 non significant wins for the models without costs and 2 non significant wins for the models with costs; ii) Regarding the Adaboost type on the same subplot, we can see 2 significant wins and 3 non significant wins for the models without cost plus 2 draws and 1 non significant win for the model variants with costs. Note that each single bar corresponds to the result of 8 Wilcoxon statistical tests, each test performed on a different data set.

This plot is quite dense and some time should be devoted to it. A first obvious observation is the abundance of yellow (draws) on the tasks with two classes. This is expected because with two classes the use of cost should only make sense if the data set is not balanced and a higher reward is given to the rarer class. Only the Airfoil and CCP data sets (see Table 2.3) have have this characteristic, which may explain the existence of a significant win with the SVM models in the Precision metric. One may also question the reason for the different behaviour of SVMs in two-class tasks. If we look at the list of models used in our experimental study (Table 2.6), we see that the SVM type is by far the one with more variants, up to a total of 32, while the other types have nearly half that amount². Naturally, when comparing 32 variants without costs with 32 with costs, the chance for the best of each group to be different from each other increases, even in tasks where cost-benefits would have little influence.

²The kernel of the SVM model has a great impact on the model itself. By choosing two types of kernel, the number of variants nearly doubles

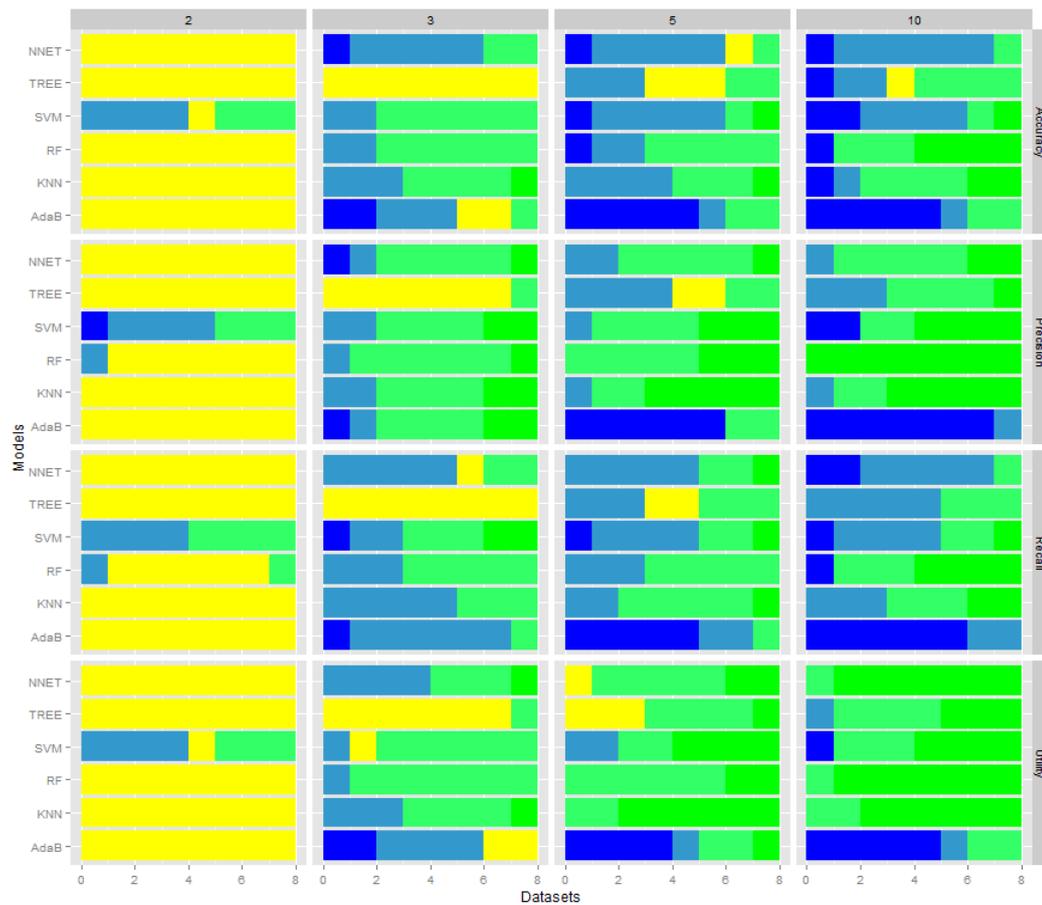


Figure 2.7: Segmented by type of model, by metric and by the number of classes of the response variable of each task, a Wilcoxon test is performed between the best model variant of each modelling tool (Classification without costs vs Classification with costs). Since there are 8 data sets in each segment, there are 8 results per segment. Each bar shows the results for each segment, where each one of the five colours is associated to a type of win (significant of the classification approach, non-significant, draw, non-significant for the regression approach and significant). The length of each colour describes the number of times that type of win occurred.

Metric by metric, we will draw some conclusions and see if they are coherent with the analysis from the previous subsection. In terms of Accuracy, it is hard to pinpoint any special advantage of any approach. There are models such as the Random Forests that seem to benefit more from the usage of costs, but we see slightly the opposite in models like the SVM and NNET and in a more severe way for the AdaBoost type with a large number of significant wins for the models not using costs. The number of classes seems to have little influence on this comparison for the accuracy metric, though there seems to be a slight improvement for the standard models for a higher number of classes. These remarks are coherent with the conclusions of the previous section, where some balance was observed, with slighter higher differences for tasks with more classes, with the exception of

the AdaBoost type, where the usage of costs decreases the Accuracy the higher the number of classes.

Concerning Precision the models with costs are clearly better across all the type of models, with the exception of Adaboost. This could be explained by the structure of the Adaboost models, since at each learning iteration of these models, every wrong prediction will be given a higher weight, somehow decreasing the severity of their mistakes. In the work of [Desai and M Jadav \(2012\)](#), we see that the standard AdaBoost models have their levels of performance decreased when cost sensitive extensions are added. In summary, with the exception of Adaboost, we have confirmed the results of the previous section where we had observed that the usage of costs-benefits improved the results in terms of Precision.

Regarding Recall we observe that the models without costs obtain better results, particularly for a higher number of classes. Still, this general trend is somewhat contradicted by KNN and Random Forests where we observe more balanced results between the two alternatives. As with Precision, this general trend of the results was already observed in the previous section.

Finally, in terms of Utility, all models except Adaboost obtain much better results when using costs. Once again, this observation is more marked in data sets with more classes.

To sum up, this second type of analysis corroborates the conclusions from the first analysis. With the exception of Adaboost we have observed the same type of effect of the usage of cost-benefit matrices on the selected metrics. In terms of Precision and Utility it is clear that the higher the number of classes, the better the models with costs will be. On the other hand, the opposite behaviour is observed for the Accuracy metric, where more classes per data set seem to favour the models without costs. This is theoretically expected since with more classes, the new models will tend to prefer forecasts of the central values/classes in order to decrease the severity of their mistakes. Naturally, this may penalise the accuracy levels, since the predicted class may not correspond anymore to the one with the highest predicted probability.

In appendix [A.1](#), we present the results of another similar experimental study. Compared to the current study, instead of comparing the best variant of each modelling technique using costs and without costs, we compare the median score of all variants of the technique. Our goal with this other experiment was to show the results of the comparison that should be expected by a user that is not willing to spend that much time looking for the best variant of each modelling algorithm. The results are similar to the ones observed using the top modelling variant per group. However, the abundance of the green colour is stronger. There are more cases in which the models with costs are now outperforming the ones without. The AdaBoost type is the most evident case, where the dominance of the blue colour is replaced by the green one. This study suggests that if a user that is not able to conduct an extensive search for the best parameters of a classification model, then by using costs they should achieve better performances.

2.5.3 Post-hoc Nemenyi test: Average Ranks

So far, we have evaluated and compared, task by task, the performance between the top modelling variants of either approach (with and without costs). This has the potential limitation of some model variant to be chosen as the best, due to some randomness (by fitting the task unusually well, where the same model variant would achieve poor performances on other tasks). This possibility raises the question if this model variant was properly representing its respective modelling approach.

Therefore, we introduce the last component of this study. The objective is to compare the best model variants of each modelling approach in terms of their overall performance across all the tasks of the same type (same number of classes), instead of conducting a separate test task by task. Hence, we may not only reinforce the quality of the conclusions drawn in the previous analysis, but also evaluate both approaches in terms of the robustness and adaptability of their model variants (since the top modelling variants will now be the ones who can perform well in several tasks at once).

In order to do that, we will use the Friedman test followed by a post-hoc Nemenyi, segmented by the number of classes of each data set. For the models without and with costs, for each set of tasks (2, 3, 5, 10 classes) and for each metric, we will rank each model variant according to its score (the best variant without costs for the accuracy metric for the Airfoil2 task receives rank 1, the second receives rank two, etc. The same is done for the models with costs). Then the procedure is repeated for the other tasks with the same number of classes, where the rankings of each model will be averaged. The five top models of each approach will be selected thus creating a new set of 10 models. Finally, the Friedman test followed by a post-hoc Nemenyi will be applied to this new set. By doing so, we are picking the best variants of each modelling approach in terms of their overall performance across all the tasks of the same type (same number of classes) and we may check whether these 5 + 5 models present similar average ranks among themselves or quite distinct ones, thus allowing to infer that one approach is performing better than the other across several tasks for a specific metric.

This methodology should become clearer after analysing Figure 2.8. Since in the 2-class tasks, the influence of the cost-benefit matrices on the predictions of the models is quite low, we start directly by analysing the tasks with three classes. As an illustrative example, let us fully analyse the Precision sub-image (the second sub-image of Figure 2.8): The labels on each side are the name of the models that are connected to an average rank value through a line. This is the average ranking that the corresponding model obtained across all the data sets with three classes. Therefore, we see that the model named as “CLASS_RF_2.v8” has achieved the best score for the Precision metric obtaining an average ranking of approximately 2. Regarding the interpretation of the name of the models, “class” stands for classification, the text after the first underscore stands for the type of model (which in this case stands for RandomForest), while the number after the

second underscore determines whether costs were applied to the model, where 1 means that no costs were used and 2 implies that costs were used to alter the predictions. Finally, if any black line is present on the sub-image, it means that we could reject the null hypothesis of the Friedman's statistical test, implying that the average ranking of all the models are not statistically equal. On the other hand, if no black lines are present then the Friedman test failed and no Nemenyi's test was performed. Lastly, the black lines also show the results of the Nemenyi's tests between every pair of models. Any pair of models that is linked by a black line has an average rank that is not significantly different according to the Nemenyi's test.

However, one may ask the meaning of the presence of a single black line that connects all the models, such as in the Accuracy sub-image. This means that the Friedman's null hypothesis was rejected but not a single Nemenyi's test was successful. This is possible because there isn't a two way equivalence between the rejection of the Friedman's null hypothesis and the rejection of the Nemenyi's null hypothesis for at least a single pair of models. Therefore, it may be possible for the Friedman's test to be verified while not having a single pair of models statistically different according to the Nemenyi's test. We take this chance to remind the reader that the second statistical test is very safe, perhaps too safe, and a proof of that is presented on the second subimage of Figure 2.8. A model that obtained an average ranking lower than 3, across 8 tasks with 10 competing models, was not considered statistically different, according to this test, from a model that almost obtained an average ranking of 7. Therefore, we should not immediately assume that there are no differences between certain model variants if this statistical test is not verified.

Having explained the details of this figure, we will now interpret the results. One should keep in mind we are merely focusing on the data sets with 3 classes. The remaining cases will be analysed later.

Regarding Accuracy, even though all the model variants are connected by the same line indicating that the Nemenyi's test failed for every combination of pairs of models, the Friedman's null hypothesis was rejected (because at least one black line is present), implying there is some significant difference across the ranking of all the models. Considering this information along with the fact that the top five average models were constructed with costs, we can state that the utilisation of cost-benefit matrices improves the overall performance in terms of accuracy of classification models, namely on 3-class tasks.

We can also observe that the models with costs are on the first five places in terms of Precision, where the top three models have a considerable distance from all the top models without costs. Surprisingly, the Nemenyi's test failed for every combination of model variants, but we know that this test is very demanding. Nevertheless, the out-performance of the new models is apparent. An interesting note is the presence of the Adaboost type on this second sub-image in contrast to the full domination of the RandomForest type across all the metrics and regardless of the utilisation of cost-benefit matrices.

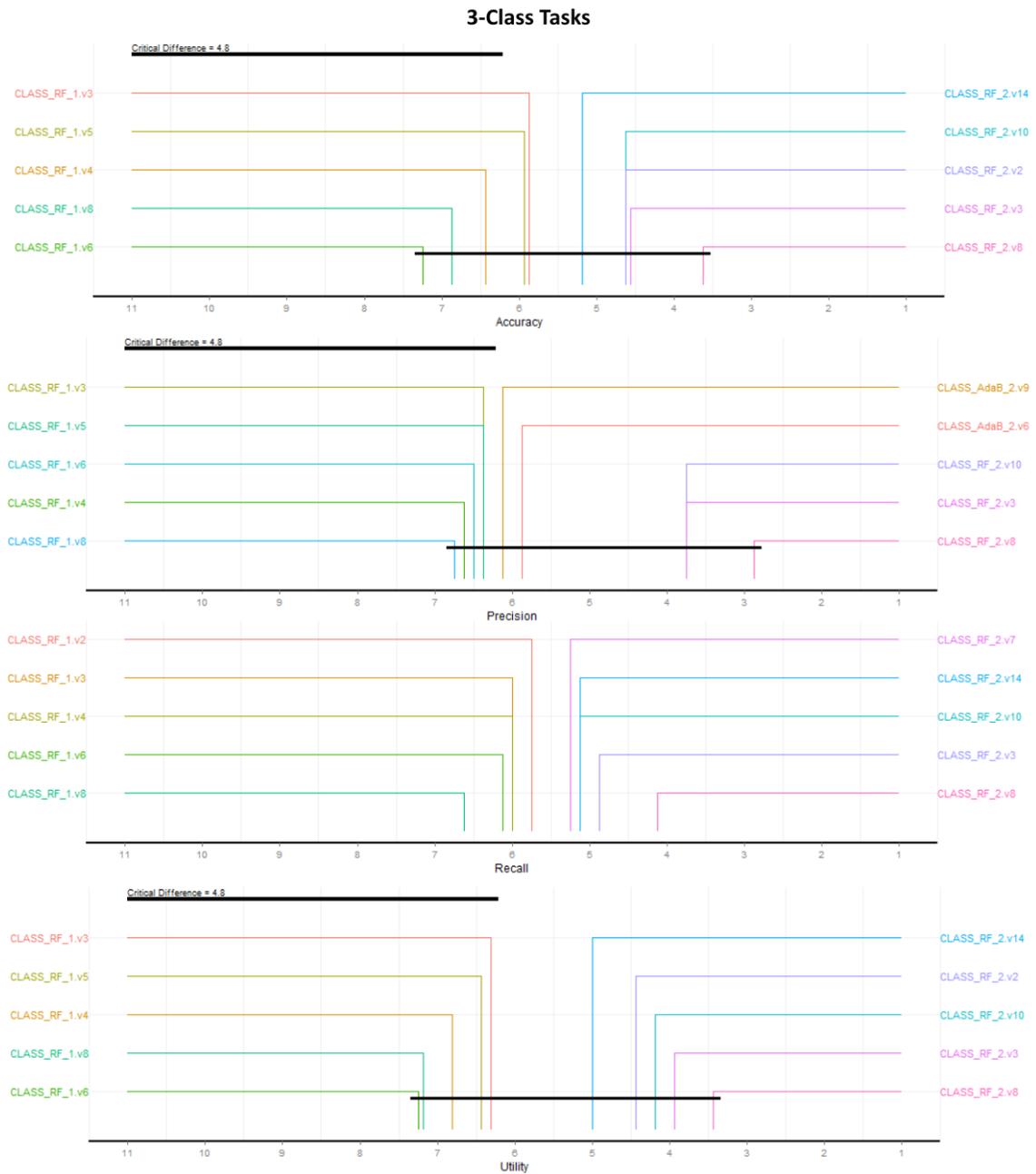


Figure 2.8: The top five average ranking model variants of both modelling approaches (Classification without costs and Classification with costs) are forming a new set of variants, and their average rankings are re-calculated within this new set of models for tasks with 3 classes only. Each model is thus given an average ranking (x axis). The presence of at least one black line implies that the Friedman’s 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’s statistical test.

Concerning Recall, despite the top five models belonging to the models with costs, not even the Friedman’s test was verified. There is no strong evidence suggesting one modelling approach to be overall outperforming the other.

Lastly, we analyse the Utility score. The top 5 is composed by all the 5 models with costs, suggesting that these models are outperforming the standard ones in this metric. Once again, only the Friedman’s null hypothesis was rejected, but combining the first test with the fact that all the top models without costs were outperformed by the new ones, there seems to exist evidence suggesting some improvement.

The lack of significant results, namely in terms of the Nemenyi’s tests, does not lower the importance of the conclusions drawn. These tests are quite demanding and only produce a p-value lower than 0.05 in very “safe” cases. Furthermore, the Friedman’s test was verified in three out of the four metrics under study, giving some strength to observed results. Summing up, in tasks whose response variable has a domain composed by 3 classes, the top five average ranking models with costs have achieved better average rankings in terms of Accuracy, Precision and Utility. The last two were expected, while the observed Accuracy results are more surprising. The models without costs are internally optimised for Accuracy, while the usage of cost-benefit matrices, as means of a post-optimisation tool, maximises the Utility score instead. Therefore, one could expect some decrease in this metric.

It is definitely interesting to observe the application of cost-benefit matrices on the standard classification models, improving the performance of these models across almost all the used metrics without compromising any other. These results are coherent with all the previous analysis, giving more robustness to our conclusions.

It is also interesting to note the complete dominance of the Random Forest models. Whether using costs or not, these models have topped every metric of performance, clearly suggesting that the best classification models to deal with the actionable forecasting problem in a more generic context is the Random Forest.

We shall now analyse the results for tasks whose response variable has 10 classes, whereas the results for 2 and 5 classes will be shown in appendices [A.2](#) and [A.3](#). With 2 classes, the differences between the use of cost and the standard models should be quite low or nonexistent, while the 5 classes case should work as an intermediate step between the 3 and 10 cases. We should expect more noticeable differences with 10 classes, since with more classes the influence of the cost-benefit matrices will certainly increase.

In [Figure 2.9](#) we can see the results for the 10-class tasks. This graph is very similar to the 3-class tasks case but with the differences between both types of models (with and without costs) more evident. In every metric, contrary to the previous case, the average rankings of some pairs of model variants are now statistically different according to the Nemenyi’s test. In the Accuracy metric, the best variant with costs is statistically different from the worst two without costs, while in the Precision one, all the top five models with costs are better than the last three of the standard variants. Moreover, the best model variant is statistically better than all the ones without costs. In the remaining two sub

images, the results are similar, with the top three model variants with costs statistically outperforming some of the variants without.

Similarly to the 3-class case, the Random Forest type completely dominated these results. It is also clear that the utilisation of cost-benefit matrices is considerably improving the performance of the classification models across several metrics, without any serious setback. This effect seems to grow with the increase of the number of classes of the target variable.

2.5.4 Conclusions

Summarising the results from the whole analysis of the usage of cost-benefits matrices to post-optimize the classification models, several conclusions can be drawn:

- Random Forest s with and without costs are constantly present in the top average models, whether with a small or large number of classes in the response variable;
- With fewer classes the Adaboost models also achieved the top average models (only with costs);
- The higher the number of classes per task, the higher the benefits of cost-benefits;
- If we were to rank the metrics that are more influenced by this feature, we would have to say that the Precision is the most favoured metric, followed by Utility. There are some advantages in terms of Accuracy with Recall being practically neutral (unless with 10 classes);
- The results are coherent across the three types of study conducted. Whether looking at the best variant of each modelling approach, or at the best variant per type of model or even the average ranking of each model individually, all the conclusions follow the same direction. The addition of costs improves the performance of the standard classification models, without any considerable disadvantage;
- The Adaboost type was the only type of model that seems to have its performance decreased by the usage of this feature.

Finally, when evaluating the Regression approach against the Classification one, we will be using all the models with and without costs on the set of classification variants. Across the several tests to be done, we should see a large set of models with and without costs being selected as the representative of that approach, with an higher incidence on the variants with costs.

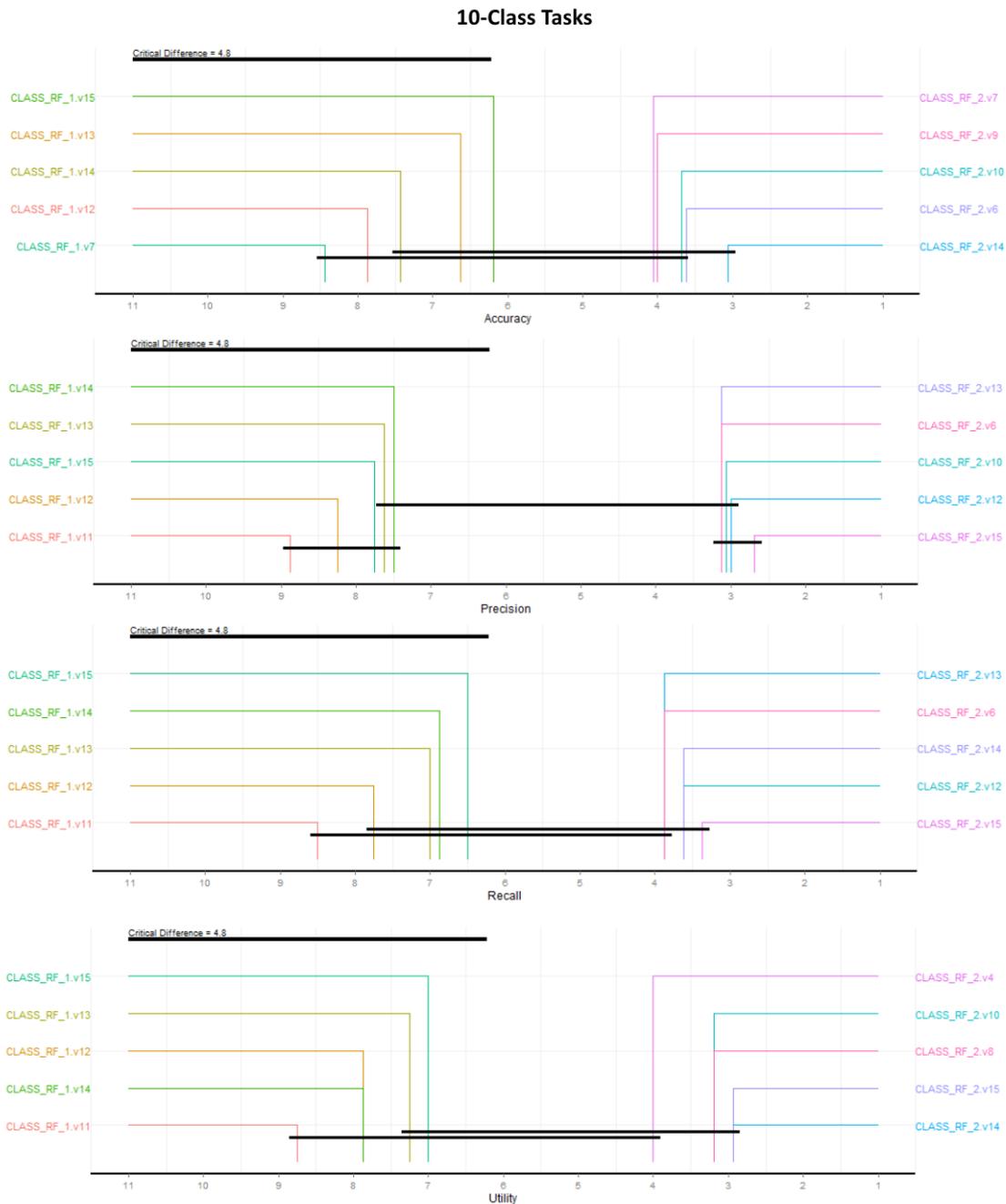


Figure 2.9: The top five average ranking model variants of both modelling approaches (Classification without costs and Classification with costs) are forming a new set of variants, and their average rankings are re-calculated within this new set of models for tasks with 10 classes only. Each model is thus given an average ranking (x axis). The presence of at least one black line implies that the Friedman's 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's statistical test.

2.6 Comparison of Regression and Classification modelling approaches

In this Section, we will finally compare the regression and classification modelling approaches to actionable forecasting. Recalling the experimental settings, we have a set of 76 classification models, which were also used with cost-benefits, totalling $76 \times 2 = 152$ different classification models. Regarding the regression approaches, we have a larger set of basic models, 97, but without any further variations. We also have 8 different problems, each pre-processed into data sets with 2, 3, 5, 10 classes in the response variable, thus leading to a total of $8 \times 4 = 32$ data sets. The experimental methodology used to compare all these approaches was described in Section 2.4.

This study will be divided in two parts (similar to the first and third parts of the analysis of the usage of cost-benefit matrices) with the goal of ensuring that our final conclusions are robust and meaningful. At first, the top modelling variant per task and for each individual metric of both classification and regressions approaches will be compared. In order to obtain stronger and statistically valid results, we will use a Wilcoxon test in every single one of those paired comparisons. Secondly, with the goal of evaluating the robustness of each modelling alternative, for every individual metric and separately for each approach, we will calculate the average ranking of each modelling variant based on their rankings per task. The top five model variants of each approach will then be selected and grouped, and a post-hoc Nemenyi statistical test will be conducted.

There are some theoretical properties of both modelling approaches that create some pre-conceptions on our expectations on the outcome of these experiments, which we plan to experimentally confirm (or not) with our results. Typically, a classification model is biased towards having the highest ratio of correct decisions (Accuracy), without having any consideration of the severity of some eventual mistakes. On the other hand, a regression model is optimised to obtain the lowest mean squared error, implying that this approach will try to forecast a numeric value close to the real one that will inevitably lead to decisions that will be at a “shorter” distance from the true value. In an actionable forecasting problem, if one is searching for the best possible ratio of correct decisions, regardless of everything else, meaning they would be only interested in the Accuracy metric, then we should see the classification modelling approach as more successful, since it is exactly what the classification models are optimised for. However, if there are some classes of decisions that are more important than others, or if there are some decisions (classes) that one may only be willing to make if there is plenty of evidence supporting it, then the regression modelling approach should become more competitive. One could even say that this approach should even outperform the classification one. However, since we also have considered a second version of all the standard classification models, these expectations may no longer be valid. Cost-benefit matrices were used with the goal of providing to the standard variants some tools to consider the severity of their mistakes, which will enhance

their performance as observed in the previous section.

2.6.1 Wilcoxon test: Best per metric and per data set

Similarly to the comparison of classification models with and without cost-benefits, we will compare and carry out a Wilcoxon statistical test between the top modelling variant of each modelling approach (classification and regression) per task. This type of comparison simulates a setup where the user is able to almost exhaustively search for the optimal parameters that will lead to the best performance possible per task. This comparison will be able to identify any characteristic that may make one modelling approach more desirable than the other.

In Figure 2.10, the results are shown for the Accuracy metric. There are 13 significant wins for the classification modelling approach against 2 of the regression one. This is an overwhelming victory, though one should keep in mind that most of the tasks that verified the Wilcoxon's test are the ones with more classes in the response variable. This suggests that most of the classification wins are occurring due to the usage of the post-optimised models with cost-benefit matrices, since with two or three classes, the number of wins falls to three cases. The regression seems to be more competitive with fewer classes

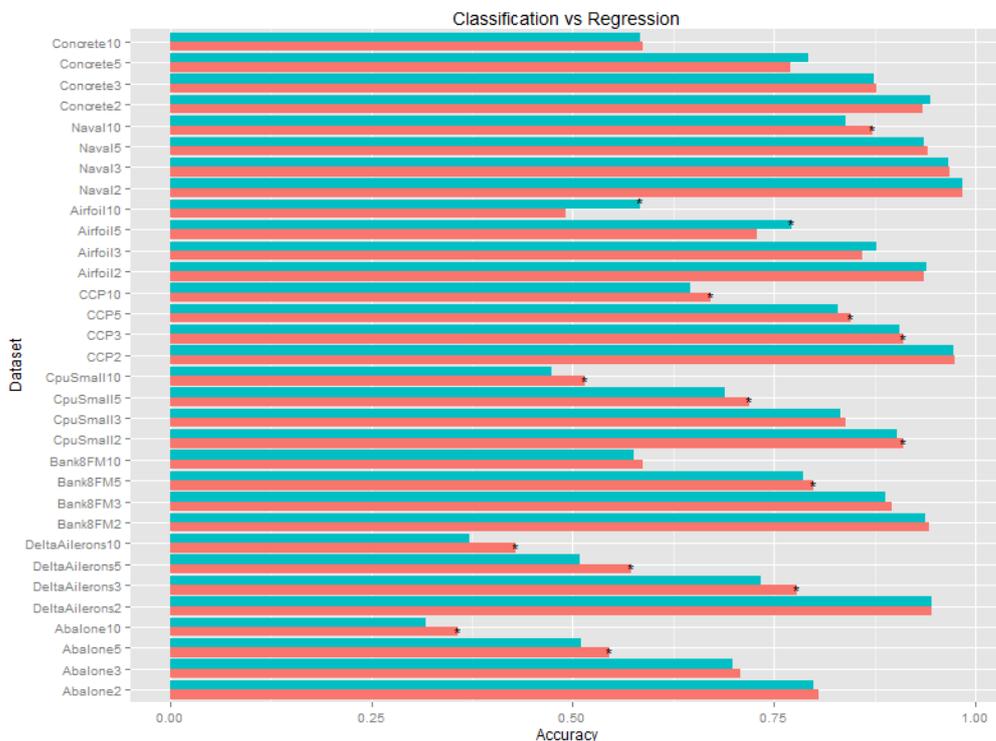


Figure 2.10: Best classification variant (red) against the best regression variant (blue) for the Accuracy score, where the presence of asterisk implies that the respective variant is significantly better, according to a Wilcoxon test with $\alpha = 0.05$.

Regarding Precision (Figure 2.11) the results appear to be quite similar to Accuracy. Another overwhelming victory of the classification modelling approach was obtained with 11 against 2 significant wins.

Figure 2.12 shows the results in terms of Recall, where we can observe that the classification modelling approach has obtained 9 significant wins against 2 of the regression models.

Therefore, it is expected that the F-Measure that equally weights Recall and Precision into a single metric will favour the Classification models, since the latter have outperformed in both cases. In Figure 2.13 we can confirm these expectations.

Lastly, when analysing the Utility score in Figure 2.14, we observe the most significant victory of the classification approach with 13 significant wins against 2 of the regression alternative.

These results provide plenty evidence that, if the users are willing to make some efforts in terms of searching the best modelling variant available in terms of tuning the parameters and testing several types of models, then they should find their best models for an actionable forecasting problem in the Classification approach, namely when using cost-benefits.

There is, however, a drawback of this type of general conclusions. By looking at each task separately, we may be disregarding some models that obtained good scores across most tasks without ever being the best model available. These models may also be of interest, since they present a stable performance across all tasks thus seeming to be able to adapt to most tasks. This motivates our second analysis that looks at the average rankings of each model.

2.6.2 Post-hoc Nemenyi test: Average Ranks

Finally, we analyse the average rankings of the best five average models per modelling approach. The main goal is to draw some conclusions regarding the robustness and adaptability that both classification or regression models may offer and determine if one approach outperforms the other in this respect.

Unlike in the analysis of cost vs no cost, we will analyse not only the tasks with 3 and 10 classes, but also the ones with 2 and 5 classes. The outcome depending on the number of classes is not obvious nor predictable, thus motivating the analysis case by case. However, we do know that by adding the model variants with costs to our set of classification models, their results were clearly improved, mainly on tasks with a higher number of classes. Therefore, the regression models should have more difficulties outperforming the classification modelling approach on data sets whose response variable has a larger domain.

In Figure 2.15, we can see the obtained results for all the tasks whose response variable is binary. The first observation is the lack of any significant results across all the metrics. However, some conclusions may still be drawn. In terms of Accuracy, Recall and Utility, the top five models (lowest average rank) belong to the classification modelling approach

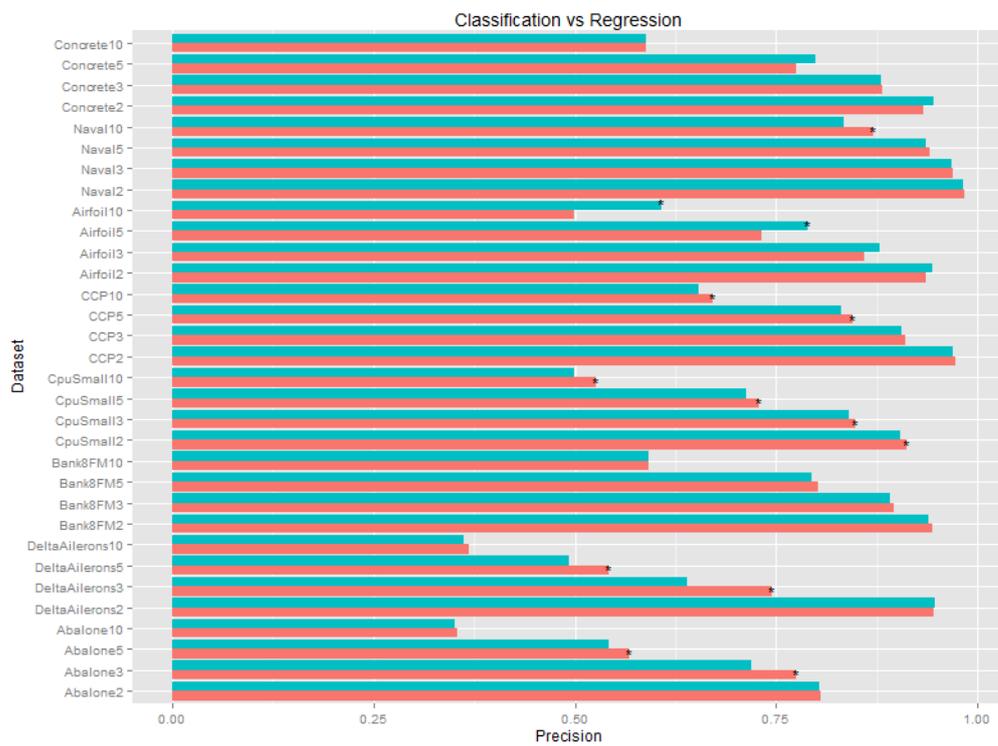


Figure 2.11: Best classification variant (red) against the best regression variant (blue) for the Precision score, where the presence of asterisk implies that the respective variant is significantly better, according to a Wilcoxon test with $\alpha = 0.05$.

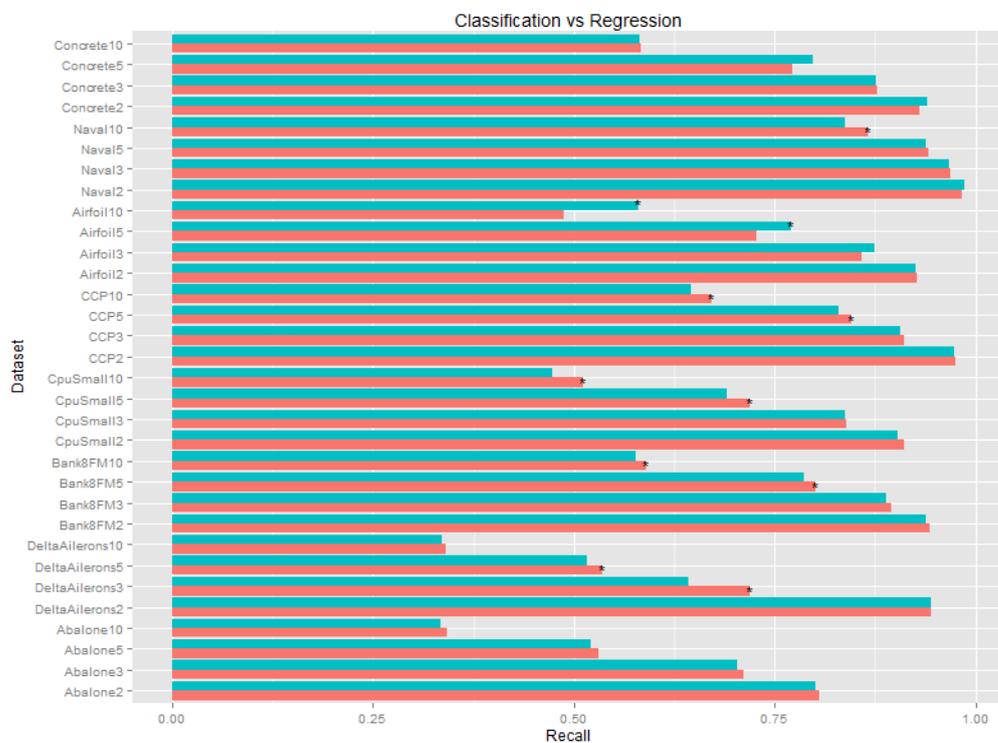


Figure 2.12: Best classification variant (red) against the best regression variant (blue) for the Recall score, where the presence of asterisk implies that the respective variant is significantly better, according to a Wilcoxon test with $\alpha = 0.05$.

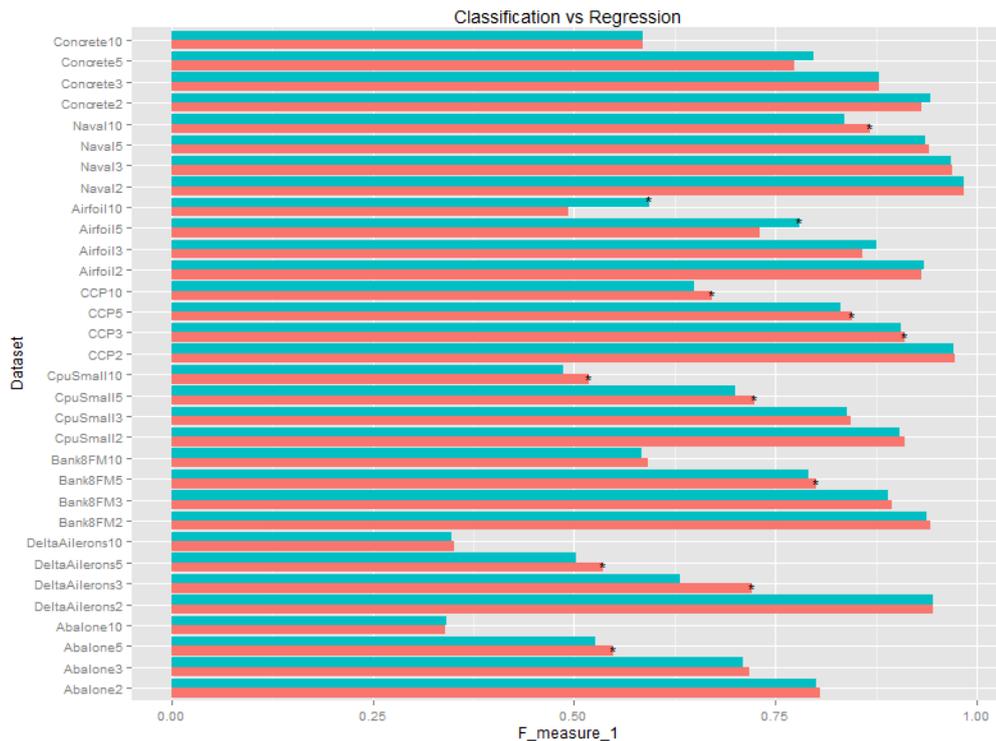


Figure 2.13: Best classification variant (red) against the best regression variant (blue) for the F-Measure score, where the presence of asterisk implies that the respective variant is significantly better, according to a Wilcoxon test with $\alpha = 0.05$.

(except the fifth position on the last metric) while in terms of Precision, the results are more even. Although the differences are small, the classification models have some advantage.

A note of interest is that the Random Forest type completely dominated the results for these tasks and most of the top classification model variants are the standard versions without the addition of cost-benefits matrices. The dominance of the Random Forest type was a constant when evaluating the classification models with and without costs, and now it seems to be the best regression modelling tool. This matter will be addressed again when analysing the remaining tasks. The last point was rather expected, since with a binary response variable, whose classes have quite similar relative frequencies, there is not much space for the cost-benefit matrices to have a serious influence.

Figure 2.16 presents the results for the 3-classes tasks. The most noticeable difference regarding the previous case is the rejection of the Friedman's null hypothesis for the Accuracy and Precision metrics. Furthermore, the average rankings of the classification models are clearly outperforming the regression ones in terms of Accuracy.

In terms of Precision, even though the Friedman's test was verified, implying that the average rankings are significantly not equal, it is not easy to pinpoint the success of one approach over the other. Indeed, the first three places are taken by classification modelling variants and with some visible distance from all the other models, but the next two models

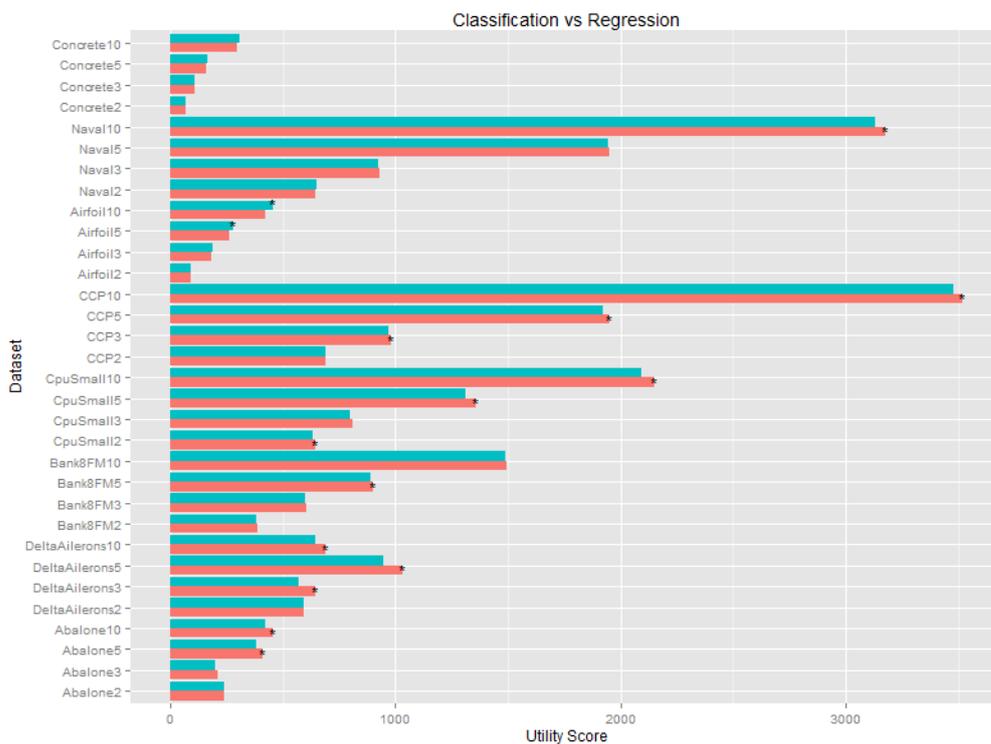


Figure 2.14: Best classification variant (red) against the best regression variant (blue) for the Utility score, where the presence of asterisk implies that the respective variant is significantly better, according to a Wilcoxon test with $\alpha = 0.05$.

belong to the regression approach. Therefore, we claim that both modelling approaches were quite even in this metric, with some slight advantage for the classification approach. Concerning the last two metrics, Recall and Utility, no significant results were obtained. Even though the top five modelling variants were all classification ones, there is not enough evidence to support any advantage of this approach. Actually, it is quite interesting to note that the regression approach is able to be competitive against models that were optimised towards the Utility score. Indeed, the process of forecasting the numeric variable is helping to reduce the severity of the errors on the predicted decisions.

Figure 2.17 shows the results for 5-classes tasks. They are quite similar to the previous case. The classification models outperform the regression ones in terms of Accuracy, but the results are more similar on the remaining metrics, with a slight advantage of the former approach in the Precision metric. Just like in the previous cases, the classification models dominate the first five positions in all the metrics, but in terms of Recall and Utility the differences are smaller.

Finally, the results on the 10-classes tasks are shown in Figure 2.18. The gap between the Accuracy score of both approaches is similar while the small advantage of the classification models on the Precision metric observed so far is strengthened. Curiously, the results regarding the Utility and Recall scores become even more balanced.

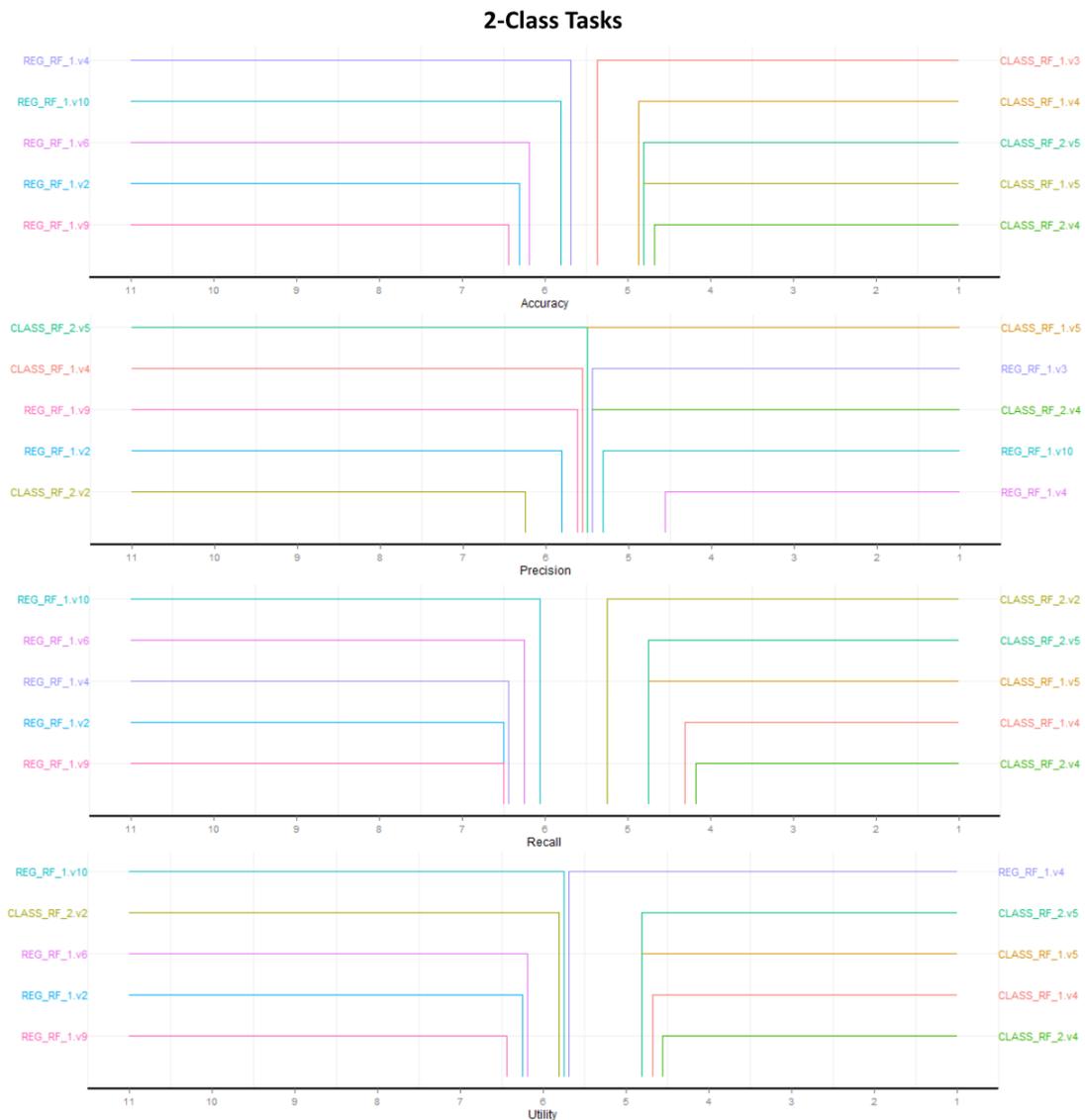


Figure 2.15: The top five average ranking model variants of both modelling approaches (Classification and Regression) are forming a new set of variants, and their average rankings are re-calculated within this new set of models for tasks with 2 classes only. Each model is thus given an average ranking (x axis). The presence of at least one black line implies that the Friedman's 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's statistical test.

Overall, the increase in the number of classes on the response variable is having little influence on the results. Actually, it seems that there is a striking difference, depending on whether the number of classes is higher than 2 or not. If it is not, then both modelling approaches seem to produce quite similar results where it is hard to pinpoint any out-performance of one over the other. This could be explained by the fact that the usage of cost-benefits has little room to significantly improve the performance of the standard

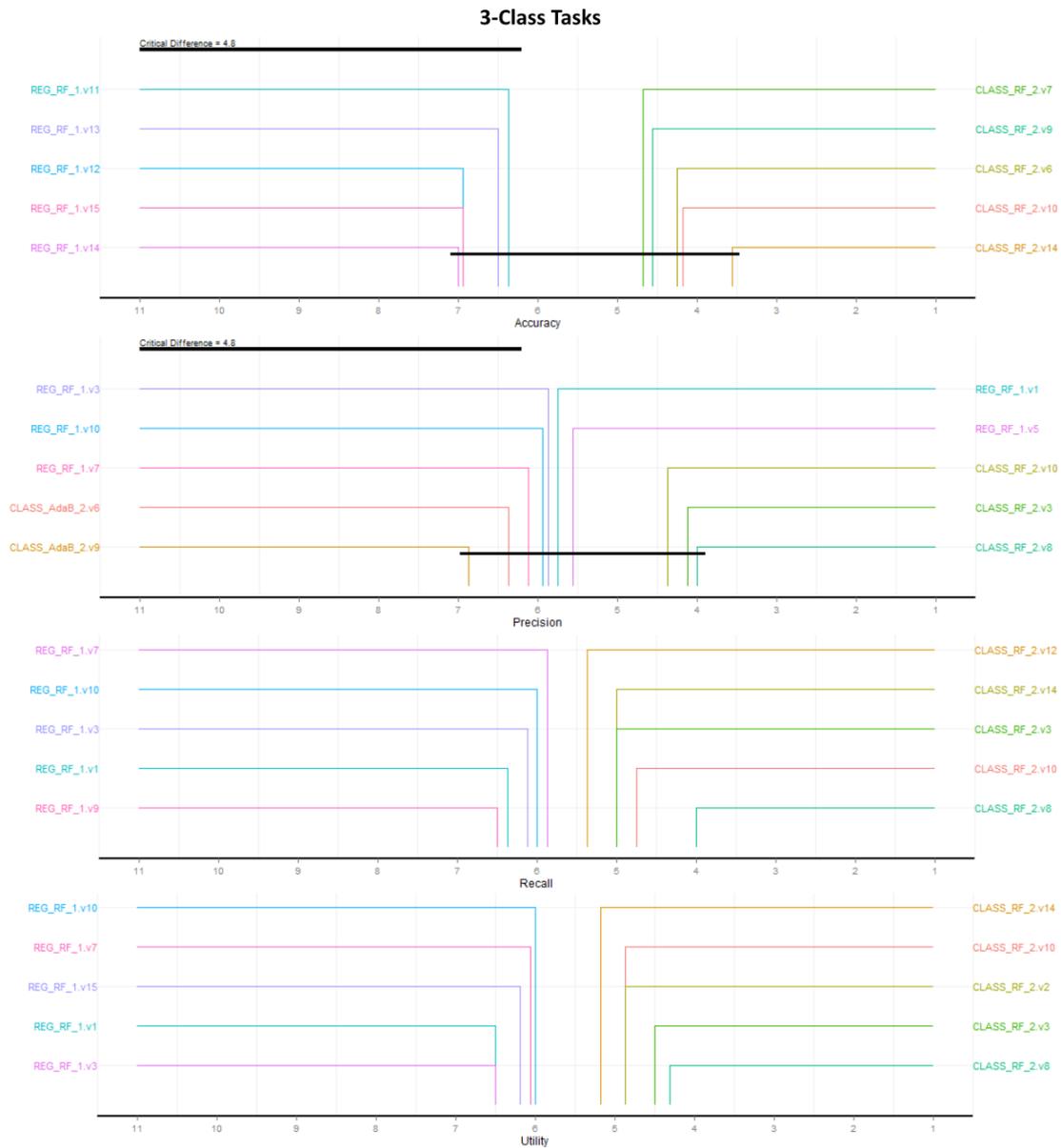


Figure 2.16: The top five average ranking model variants of both modelling approaches (Classification and Regression) are forming a new set of variants, and their average rankings are re-calculated within this new set of models for tasks with 3 classes only. Each model is thus given an average ranking (x axis). The presence of at least one black line implies that the Friedman's 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's statistical test.

models. With 3 or more classes, a gap between the Accuracy of both approaches appears, and is kept constant regardless of the number of classes. In terms of Precision, a slight advantage for the classification models was observed, that is also kept fairly constant across the different number of classes. However, when evaluating the Recall and Utility of both approaches, it seems that the increase of the number of classes (decisions) makes both

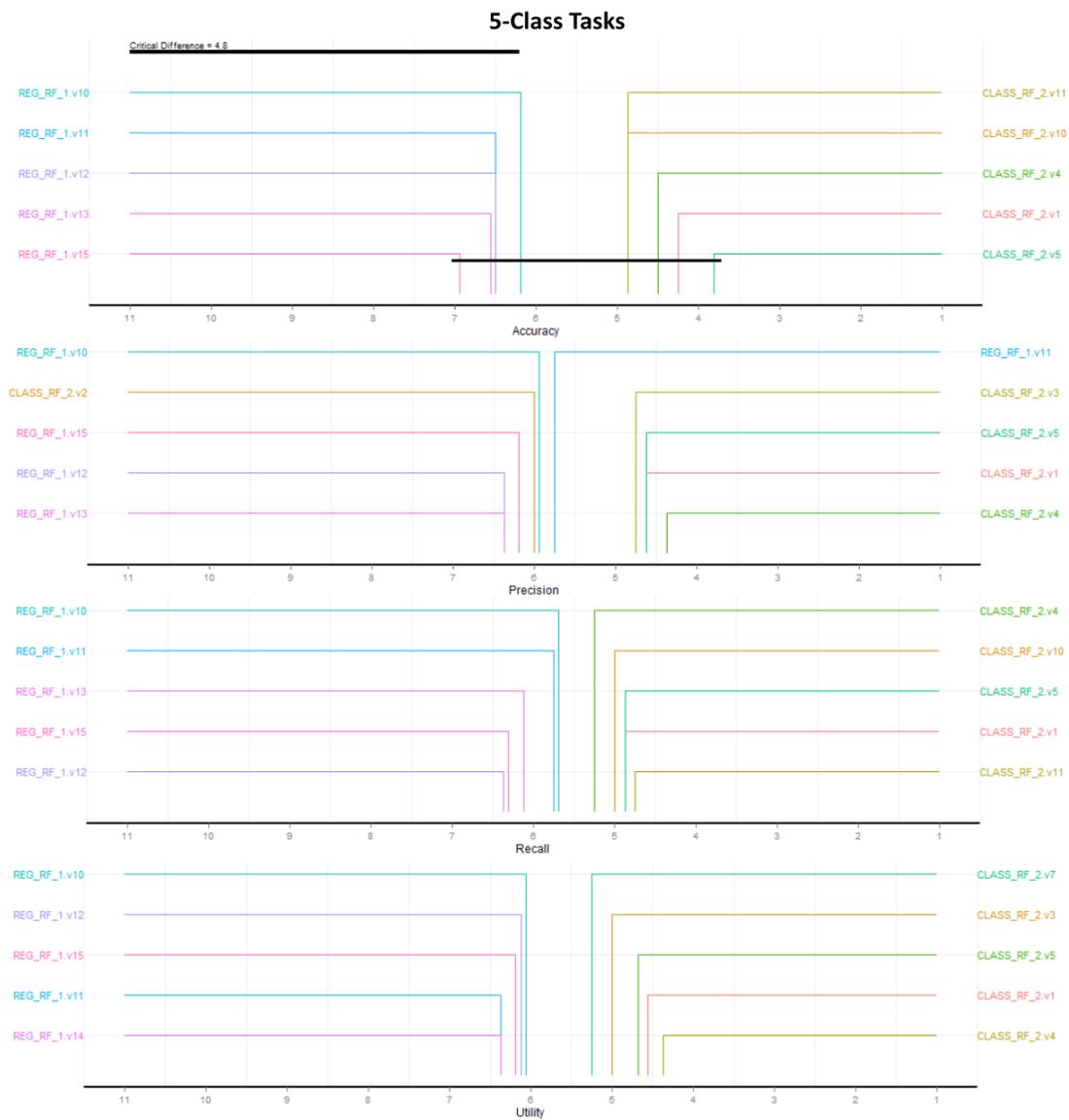


Figure 2.17: The top five average ranking model variants of both modelling approaches (Classification and Regression) are forming a new set of variants, and their average rankings are re-calculated within this new set of models for tasks with 5 classes only. Each model is thus given an average ranking (x axis). The presence of at least one black line implies that the Friedman's 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's statistical test.

approaches more and more balanced.

The results from both analysis (top modelling variant per task and this last one) are coherent. The first has produced strong results suggesting an advantage of the classification modelling approach. In the latter, even though the regression approach was able to become more competitive in some specific cases (for instance when considering the Recall or the

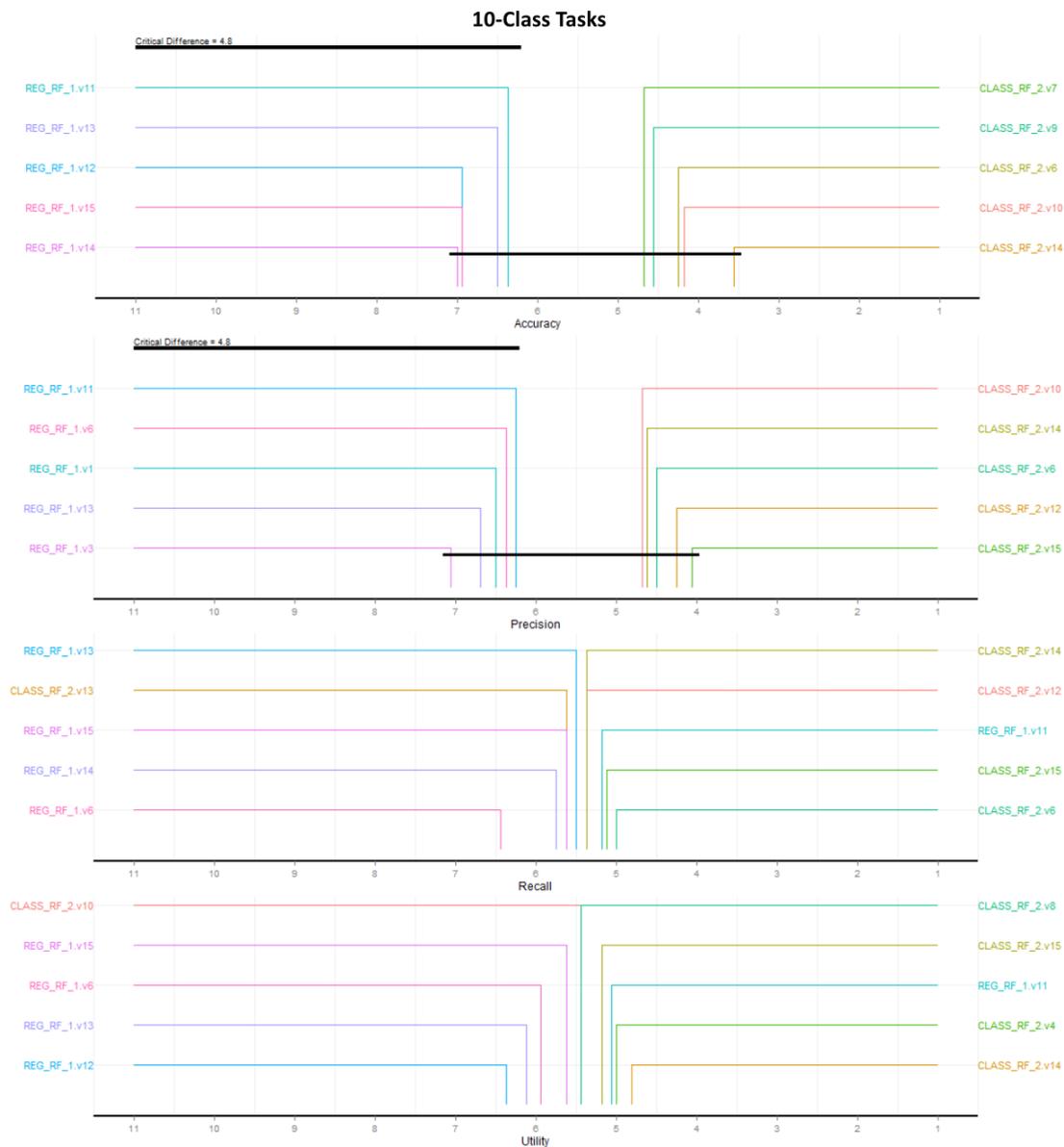


Figure 2.18: The top five average ranking model variants of both modelling approaches (Classification and Regression) are forming a new set of variants, and their average rankings are re-calculated within this new set of models for tasks with 10 classes only. Each model is thus given an average ranking (x axis). The presence of at least one black line implies that the Friedman's 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's statistical test.

Utility for tasks with more classes), the best average ranking model variants were always belonging to the classification approach. Both studies point in the same direction from different perspectives, strengthening the validity of our conclusions.

2.7 Conclusions

This chapter has presented a comparative study of two different approaches to deal with the actionable forecasting problem. The first, and more conventional approach, uses regression tools to forecast the unknown numeric value and then uses some decision rules to choose the "correct" decision based on these predictions. The second approach tries to directly forecast the "correct" decision. We have compared both approaches on a diverse set of generic tasks. We have also used an extensive set of modelling tools, considering a large set of parameter variants for each one, in order to guarantee some robustness of our conclusions.

Overall, for these generic actionable forecasting tasks, we have observed some consistent results that we now summarise. If the user is willing to perform an extensive search of the best model types and parameter variants for each specific task, then for most tasks the best results were obtained using the classification modelling approach. If extensive model tuning is not an hypothesis for the user then our experiments lead to two main conclusions. If the goal consists solely on reaching the best possible ratio of correct decisions (i.e. Accuracy), then a classification tool should be selected, mainly in tasks whose response variable domain has at least 3 classes (decisions). On the other hand, one may want to consider the severity of the mistakes associated to some prediction, or even consider that the possible actions/decisions to make are not all equally important. In that case, both modelling approaches seem to be quite competitive against each other, regardless of the number of classes of the response variable. Given the large set of classification and regression models that were considered, as well as different approaches to the learning task, we claim that there is significant experimental evidence to support these conclusions.

The experiments carried out in this chapter have also allowed us to draw some other conclusions in terms of the use of cost-benefit matrices. In a generic context, as in this analysis, the use of cost-benefit matrices as an effort to maximise the utility of the predictions of the models, has clearly enhanced the performance on some metrics for a high percentage of modelling variants, without seriously compromising any other metric. Knowing that the models with costs obtained several significant wins over the models not using them, mainly on tasks with more classes, and considering that the regression modelling approach has managed to be competitive across all the tasks, we could say that the regression approach should outperform the classification approach, mainly on tasks with more classes, if classification models were not used with costs.

Chapter 3

An Application to Financial Trading

The concept of actionable forecasting appears in many domains. In the previous chapter two approaches to deal with this problem were presented and experimentally evaluated on a large set of generic tasks. In this chapter we focus on one particular application where this concept is of great relevance: financial trading. Machine Learning models are widely used to predict the evolution of financial markets. Having made a prediction on the future value of some assets, one can decide to either open a long/short position or hold. This problem fits perfectly the structure of actionable forecasting tasks and given that a considerable percentage of nowadays trades are made making use of Machine Learning models and also that large amounts of money are usually involved in this process (and therefore, very risky decisions), then there is enough motivation to address this specific task in detail. Compared to the tasks in the previous chapter there is a key difference in financial trading. Contrary to the previous problems, here data is time dependent, so different procedures must be considered in several aspects of our comparative studies.

We describe the trading problem according to the problem formalisation of actionable forecasting described in Section 2.1 of the previous chapter. The predictors \mathbf{x} 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. The g function will be defined in the next section. Similarly to the generic tasks, a regression model may be used to approximate the function $f(\mathbf{x})$, and a deterministic mapping function could then be used to produce a decision. Once again, as an alternative, a classification model could be used to directly approximate $g(f(\mathbf{x}))$.

As mentioned above our predictive tasks will be temporal in contrast with the tasks used in the previous chapter. This fact has an impact on the experimental methodology (that allows the production of reliable score estimates) used to compare different

approaches. Furthermore, we will be facing unbalanced response variables (as the number of observations that lead to the opening of long/short positions is typically much lower than the number of times that one should hold his position). Given all these differences, there may be differences in the conclusions regarding the comparison among the two alternative approaches to actionable forecasting that we have reached in the previous chapter. Checking this hypothesis is one of the goals of the work described in this chapter.

3.1 Material and Methods

This section describes the main issues involved in the experiments we will carry out with the goal of comparing the two possible approaches described in the previous chapter in the context of financial trading. The way that both regression and classification modelling approaches are applied is the same as in the previous generic study. While the first forecasts a numeric variable that will then be used as an input of a deterministic g function, responsible for mapping that forecast into a final decision, the latter will directly forecast the decisions.

The tasks as well as the g function will be described in the following subsection.

3.1.1 The Task

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. We will be working with a one-day forecasting horizon, i.e. take a decision based on the forecasts of the assets variation for one day ahead. Moreover, we will be working exclusively with the closing prices of each trading session, i.e. 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. If we forecast an increase above 2% of the closing prices our decision will be to Buy the assets, whilst a prediction of a 2% decrease in the prices will lead to a Sell decisions. Any other forecast leads to the Hold decisions

(i.e. no trading is carried out). 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’s 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. As an illustrating example, in Table 3.1, a small sample of the constructed data set regarding the Apple shares is shown. The first two columns are the response variable for the classification and regression models respectively, while the remaining correspond to subset of the used predictors. $Diff_x$ correspond to the x -th past daily variations while $Delt_x$ to the total variation between the past observation the x -th past one. The ATR column is the Welles Wilder’s style moving average, Vol is the annual volatility based on the past 10 observations and SAR is the stop and reverse point developed by J. Welles Wilder. Other predictors such as the mean of the past observations as well as different values of x for the $Diff$ and $Delt$ variables were also considered.

	$A = g(Y)$	$Y = f(\mathbf{x})$	$Diff_1$	$Diff_2$	$Delt_2$	$Delt_3$	ATR	Vol	SAR
1981-01-06	s	-0.04	-0.04	-0.02	-0.07	-0.06	1.33	0.04	30.55
1981-01-07	s	-0.02	-0.04	-0.04	-0.09	-0.10	1.34	0.04	30.88
1981-01-08	b	0.05	-0.02	-0.04	-0.06	-0.10	1.29	0.04	36.13
1981-01-09	h	-0.01	0.05	-0.02	0.03	-0.01	1.32	0.04	36.13
1981-01-12	s	-0.04	-0.01	0.05	0.05	0.02	1.24	0.04	35.89

Table 3.1: Sample of pre-processed Apple stocks variations for one-day forecasting horizon.

At this stage it is important to remark that the prediction tasks we are facing have some characteristics that turn them into particularly challenging tasks. 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 that will be provided 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 modelling techniques. Later, we will describe some of the measures we have taken to alleviate this problem.

3.1.2 Evaluation Metrics

In the previous chapter, generic classification metrics were used to compare the two alternative approaches to actionable forecasting. Whilst they can give some solid information about the value of each model, the ultimate goal of any evaluation procedure is to measure the value of the models in the context of the domain preference criteria. Therefore, 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. To make our experiments more realistic we will consider a transaction cost of 2% for each Buy or Sell decision a model may take. The Total Return was chosen over the Average Return due to the presence of models that by predicting a very small set of opening signals could achieve a quite high average score, and yet being financially uninteresting.

However, when we feel necessary, we may use some of the classification metrics presented in the previous chapter in order to obtain additional information of the behaviour of the models being tested, such as the macro-recall of the rare signals (sell and buy signals) to test the model's ability to detect those rare events that may lead to some potential returns, or the macro-precision of the same rare classes in order to test the model's confidence when making that prediction, since incorrectly forecasting those classes may cause large losses of money. These metrics were described in Section 2.3.2.

3.1.3 The Models

The predictive tasks we are facing have two main difficulties: (i) the fact that the distribution of the target variables is highly imbalanced, with the more relevant values being less frequent; and (ii) the fact that there is an implicit ordering among the decisions. The first problem causes most modelling 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. This problem was also present and worked in the previous chapter.

These two problems lead us to consider several alternatives to our base modelling approaches already used in the previous chapter and described in Tables 2.5 and 2.6 (c.f. page 15). 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 et al., 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 et al. (2013), 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 this issue (that was also considered in the generic study). Namely, we have used a cost-benefit matrix that allows us 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 maximises the utility instead of the class that has the highest probability. In the trading problem we can choose a more specific and realistic cost-benefit matrix, since each action directly leads to a profit, loss or to a missed chance of obtaining a profit.

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 as well as the hypothesis that using cost-benefit matrices would enhance the performance of the different classification models for the same tasks. The results will be shown after describing the experimental methodology.

3.2 The Experimental Methodology

Given the temporal component of the trading tasks described in this chapter, we can not use the 10-fold Cross Validation methodology to estimate the performance of a certain model as in the generic study conducted before. This procedure assumes that the data has no order including reshuffling steps, and by using it we would obtain unreliable estimates. In this context, we have decided to use a Monte Carlo simulation method for obtaining our estimates. This methodology consists 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 sub-sequent 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, as in the previous chapter, the recommendations of the work by [Demšar \(2006\)](#). The way we obtain our estimated scores is not the same as in the generic study, but the analysis of the differences of the score can be conducted in the same way. More specifically, in situations where we are comparing k alternative models on one specific task we have used the Wilcoxon signed rank test to test 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.

3.3 Hypothesis testing

With the experimental methodology described, we can finally test our proposed hypothesis regarding the improvement of our base models using a specific re-sampling method on both modelling approaches and cost-benefit matrices for the classification approach only. This

analysis will be similar to the one carried out in Section 2.5 of the previous chapter, but with three cases instead of one: i) Usage of SMOTE on classification models; ii) Usage of SMOTE on regression models; iii) Usage of cost-benefit matrices on classification models.

3.3.1 Hypothesis 1: Re-sampling the data sets - Classification

In this section we test our first hypothesis that states that re-sampling the data sets with the goal of balancing the response variable will enhance the performance of classification models. We start by comparing the top modelling variant obtained without using re-sampling against the best one using SMOTE, company by company, applying a Wilcoxon test in each case. In Figure 3.1 we analyse recall and precision for the buy and sell signals combined. The results were somewhat expected. The models applied to the data set with re-sampling achieved significantly better results in terms of recall and worse in terms of precision. This means that this implementation is making the models more capable of detecting the rare classes, which makes sense since the respective training data has more buy and sell signals. However, this increase of recall comes at the cost of making more mistakes, leading thus to worse results in terms of precision. Since we are in a trading market context, we should prefer safer decisions rather than riskier ones. Without checking the impact in terms of financial results, we can not yet conclude if the usage of SMOTE is advantageous in our problem, but these results could still be interesting in a more general problem, where detecting the rare class may be of an extreme importance, such as diagnosis problems.



Figure 3.1: 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 3.2 allows us to analyse the financial consequences of the re-sampling procedure. We can observe that in terms of Total Return and Sharpe Ratio the models applied to data sets without any re-sampling achieved significantly better results and, in several cases, by a large margin. This means that the results of pre-processing the data using re-sampling is leading to models that make more risky decisions and that these decisions are often wrong, leading to catastrophic losses.

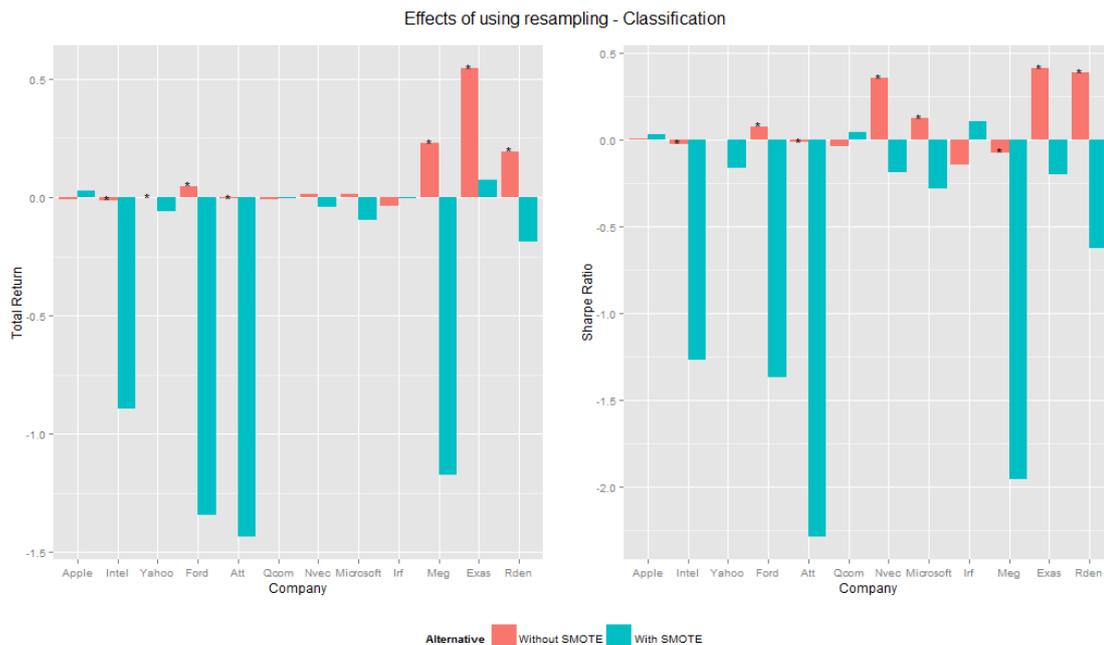


Figure 3.2: Best classification variant without SMOTE against the best classification with SMOTE for the Total Return and Annualised Sharpe Ratio metrics (asterisks denote that the respective variant is significantly better, according to a Wilcoxon test with $\alpha = 0.05$).

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 we are looking at the best variant of each type (with and without SMOTE). Although we may be tempted to think that the behaviour may happen with all the models overall, we have no solid evidence to state that. Therefore, we will test this hypothesis in another way. Similarly to the analysis of the usage of cost-benefit matrices for generic tasks, we will group the variants according to the type of model and analyse the impact of using SMOTE per type of model. In other words, we will compare the SVM variants applied to data sets without re-sampling against the same SVM variants where the SMOTE was applied. The procedure is then repeated for the other type of models. We will compare the top modelling variant of each group as well as the median score (per group).

Regarding the top modelling variant of each type, the results are shown in Figure 3.3. The first thing to notice is that the recall was significantly better every single time across all the type of models where re-sampling was applied. On the other way, the precision was worse almost every time, except on SVM's. There is no doubt that the use of the SMOTE algorithm is making the models more capable of detecting the buy and sell classes at the cost of riskier decisions. Concerning the financial metrics, we observe a clear advantage of the standard models (i.e. no re-sampling applied), with SVM and AdaBoost being the only models benefiting from the use of re-sampling on some tasks.

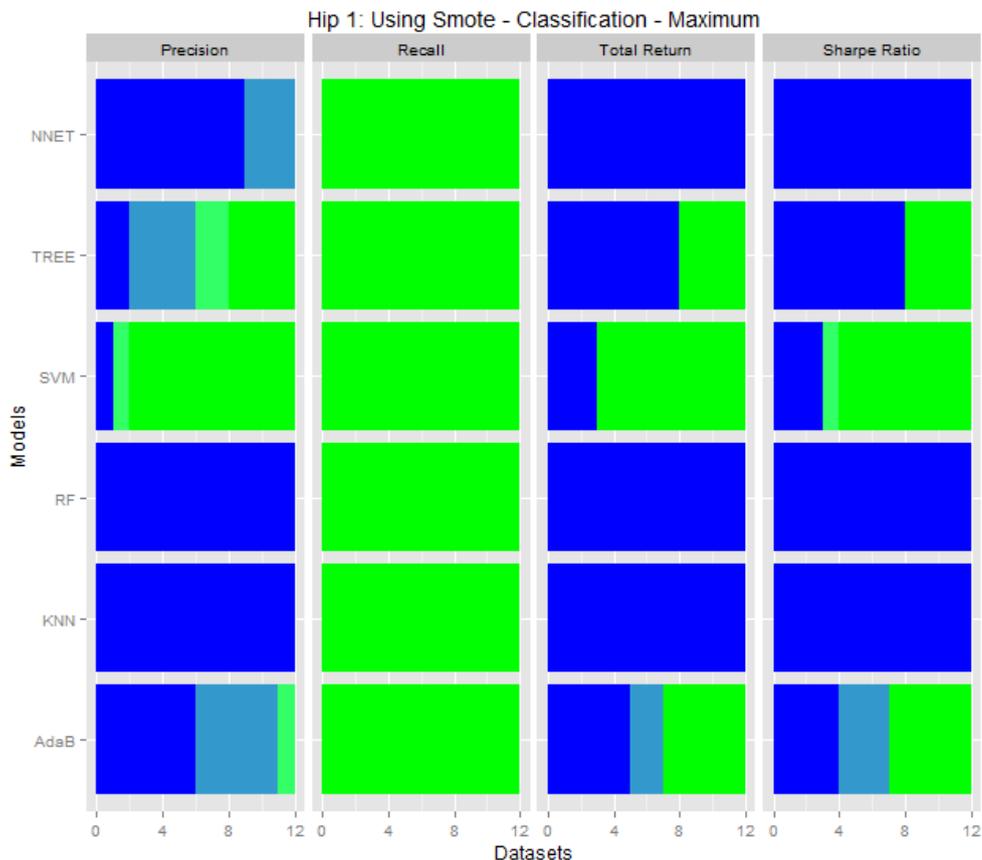


Figure 3.3: Segmented by type of model and by metric, a Wilcoxon test is performed between the best model variant of each modelling tool (Classification without SMOTE vs Classification with SMOTE). Since there are 12 data sets in each segment, then there are 12 results per segment. Each bar shows the results for each segment, where each one of the five colours is associated to a type of win (significant/non-significant win without SMOTE - strong/light blue, draw - yellow, non-significant/significant win with SMOTE - light/strong green) The length of each colour describes the number of times that type of win occurred.

Overall, the conclusions from this study are similar to the ones from the first analysis. However, in both cases (analysing all types together or each individual type), we have looked to the top modelling variant, meaning that this behaviour may not be representative of all model variants for each type of model. In order to study this spectrum of variants,

we will do a very similar study, but instead of considering the best variant of each type of model, we will consider the median score of all the variants inside that same group. Note that we no longer can use statistical tests, because we are not selecting a variant to be the representative of a group, but rather the median of the score of all the respective model variants.

The results are presented in Appendix B.1. The conclusions are very similar to the ones using the best variant. There are some subtle differences though, namely on the financial metrics. The standard models were better almost every single time. This is an interesting observation, since there were twotypes of models that were benefiting from the re-sampling on some tasks when considering their top modelling variant. It seems that if one is not willing to do extensive model tuning in this type of trading tasks, then the usage of SMOTE is not to be recommended.

Finally, we have also calculated, for every individual metric, the average rankings of each model across all the companies, making use of a post-hoc Nemenyi's statistical test. By doing so, we can compare the most robust and adaptive modelling variants with and without the usage of SMOTE, since the best models from this study will be the ones that could perform well across all the companies.

The results are shown in Figure 3.4. The model names with a “_1” are related to the standard versions where no SMOTE was applied, while the ones with “_3” were obtained using this re-sampling method. The conclusions are consistent with the previous parts of this analysis. While in terms of Macro-Recall (of the buy and sell signals), the models created using SMOTE can statistically outperform the others, the opposite happens in a clear way for all the remaining metrics.

To sum up, we have collected sufficient evidence to conclude that re-sampling the data sets for the classification models will have a negative impact on their performance, namely when considering trading evaluation metrics. This behaviour was observed both when evaluating the top variant per task, the top variant per type model, the median score within each type of model, or the average rankings across all the metrics. Only the SVM, TREE and AdaBoost types of models have benefited from the re-sampling method on a small set of tasks. We will now carry out a similar study on the impact of re-sampling regarding the regression modelling approach.

3.3.2 Hypothesis 1: Re-sampling the data sets - Regression

Once again we will start by analysing the top modelling variant of all the models without re-sampling against the top one using SMOTE. The results are shown in Figure 3.5, where we see a very similar behaviour as in the classification case regarding the recall and precision of the buy and sell signals. As before, when the re-sampling method was used, the models obtained a significantly better recall but again at the cost of making more mistakes thus leading to poor precision results.

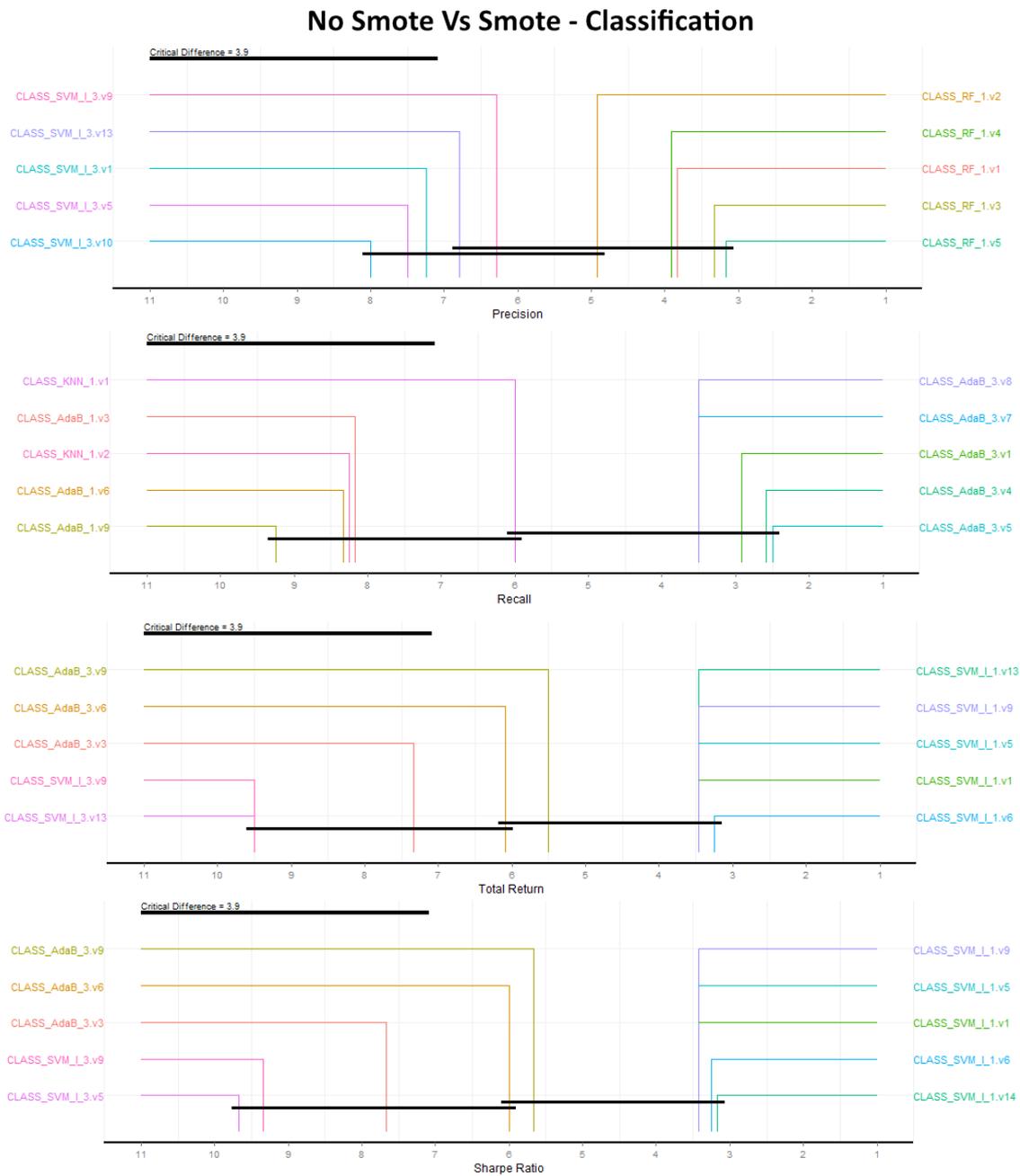


Figure 3.4: The top five average ranking model variants of both modelling alternatives (Classification with and without the usage of SMOTE) are forming a new set of variants, and their average rankings are re-calculated. Each model is thus given an average ranking (x axis). The presence of at least one black line implies that the Friedman’s 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’s statistical test.

In Figure 3.6 we can check the corresponding financial results in terms of Total Return and Sharpe Ratio scores for regression models with and without the usage of SMOTE. Even though we observe that not using the re-sampling method leads to better results in every single case, the differences are definitely smaller than in the classification case.



Figure 3.5: Best regression 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 $\alpha = 0.05$).

This may be explained by the way we create artificial examples of the minority classes. In the regression data sets, when a case is created it is not grouped into a class. This may be a considerable advantage. Every single artificial observation (created by the SMOTE algorithm) is composed by a set of the artificially assigned values of the predictors and the respective value of the response variable. It is possible that for the given set of those explanatory values, the respective true response value would be different from the one created by the SMOTE method. While in the regression approach, this error is numeric and it is expected to be small, in the classification approach it may lead to a whole different class, thus eventually confusing more seriously the respective models.

Proceeding as before, we will now look at the top modelling variants of each type of model. The results are given in Figure 3.7. In terms of Recall of the combined minority classes, the models with SMOTE were always better (or at least not worse), except one single time. However, there are several cases that are not statistically significant according to the Wilcoxon statistical test. Regarding Precision, the standard models are overall better, though the Neural Networks and Tree types did not verify this trend. Finally, considering the financial metrics all the models but the NNET, TREE and KNN were in-arguably better without the usage of re-sampling, while the others presented even results.

Finally, we also considered the median score of each group in Appendix B.2. Similar behaviours are observed, with the main difference being the fact that the median score of the standard KNN and NNET models on the financial metrics out-performed the respective variants with SMOTE. If one is not willing to make an exhaustive search for the ideal

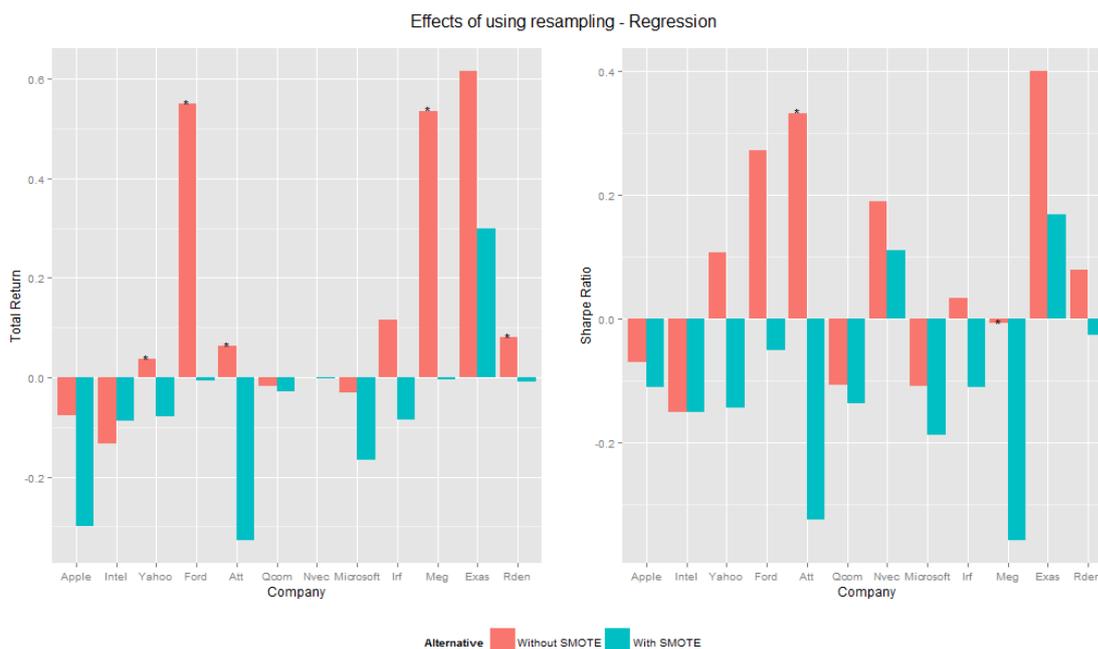


Figure 3.6: Best regression variant without SMOTE against the best regression with SMOTE for the Total Return and Annualised Sharpe Ratio metrics (asterisks denote that the respective variant is significantly better, according to a Wilcoxon test with $\alpha = 0.05$).

parameters, then the usage of SMOTE is not advisable at all.

Similarly to the testing of SMOTE on the classification models, we also consider the average rankings across all the companies making use of a post-hoc Nemenyi's statistical test. The results are given in Figure 3.8. The model names with a “_1” are related to the standard versions where no SMOTE was applied, while the ones with “_2” were constructed using this re-sampling method. The observed scores in terms of Recall and Precision are coherent with what we have seen before. The usage of resampling significantly boost the Recall level at the cost of a severe decrease in terms of Precision. Regarding the trading metrics, the results are surprisingly more even, though the standard modelling variants were always superior.

In summary, our experiments provide strong empirical evidence that using SMOTE to try to overcome the limitation of few existing extreme variations on the prices is not improving the results of the resulting models. Equivalently to the classification case, using the re-sampling algorithm increases the number of buy and sell signals forecasted by the regression models, but at the cost of incorporating too much risk. Therefore, in terms of the trading metrics, the best results across all the types of analysis are obtained when not using SMOTE. However, the TREE and KNN types of model could actually benefit from this feature on some tasks.

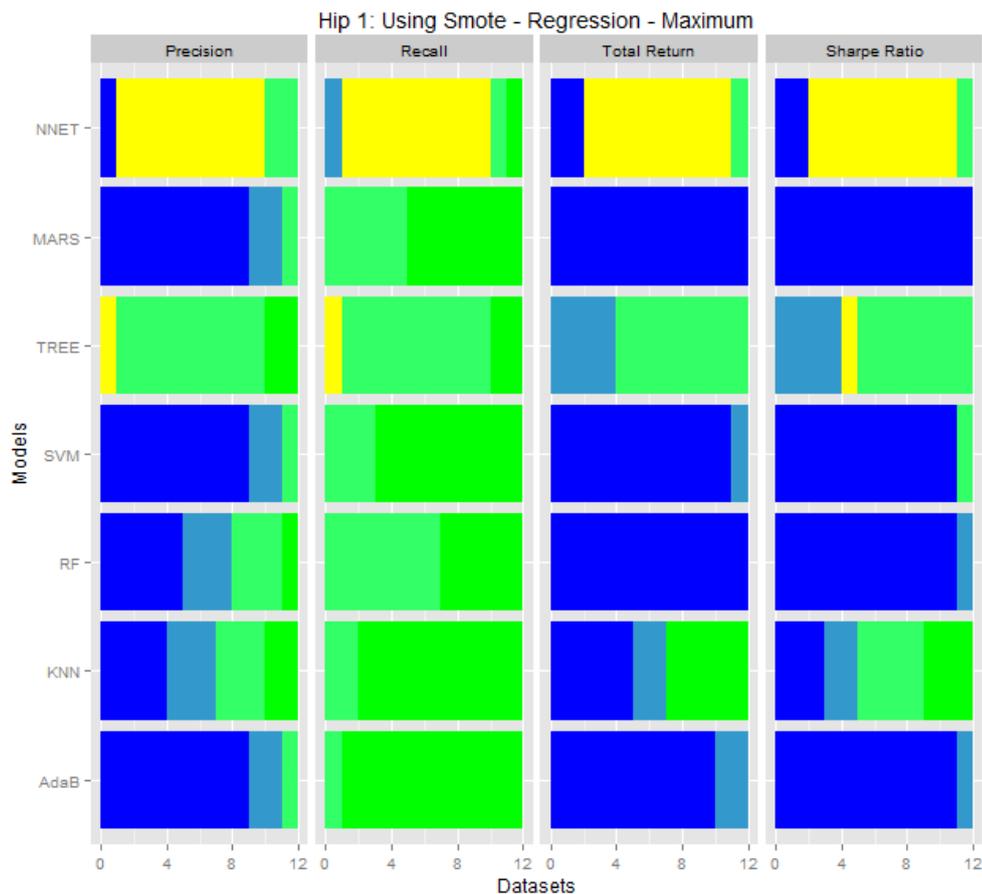


Figure 3.7: Segmented by type of model and by metric, a Wilcoxon test is performed between the best model variant of each modelling tool (Regression without SMOTE vs Regression with SMOTE). Since there are 12 data sets in each segment, then there are 12 results per segment. Each bar shows the results for each segment, where each one of the five colours is associated to a type of win (significant/non-significant win without SMOTE - strong/light blue, draw - yellow, non-significant/significant win with SMOTE - light/strong green) The length of each colour describes the number of times that type of win occurred.

3.3.3 Hypothesis 2: Adding cost-benefits

This section considers the second hypothesis we have put forward, namely that the usage of cost-benefits matrices will boost the performance of the classification models, since the information on the implicit ordering among the classes will be passed to the models.

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

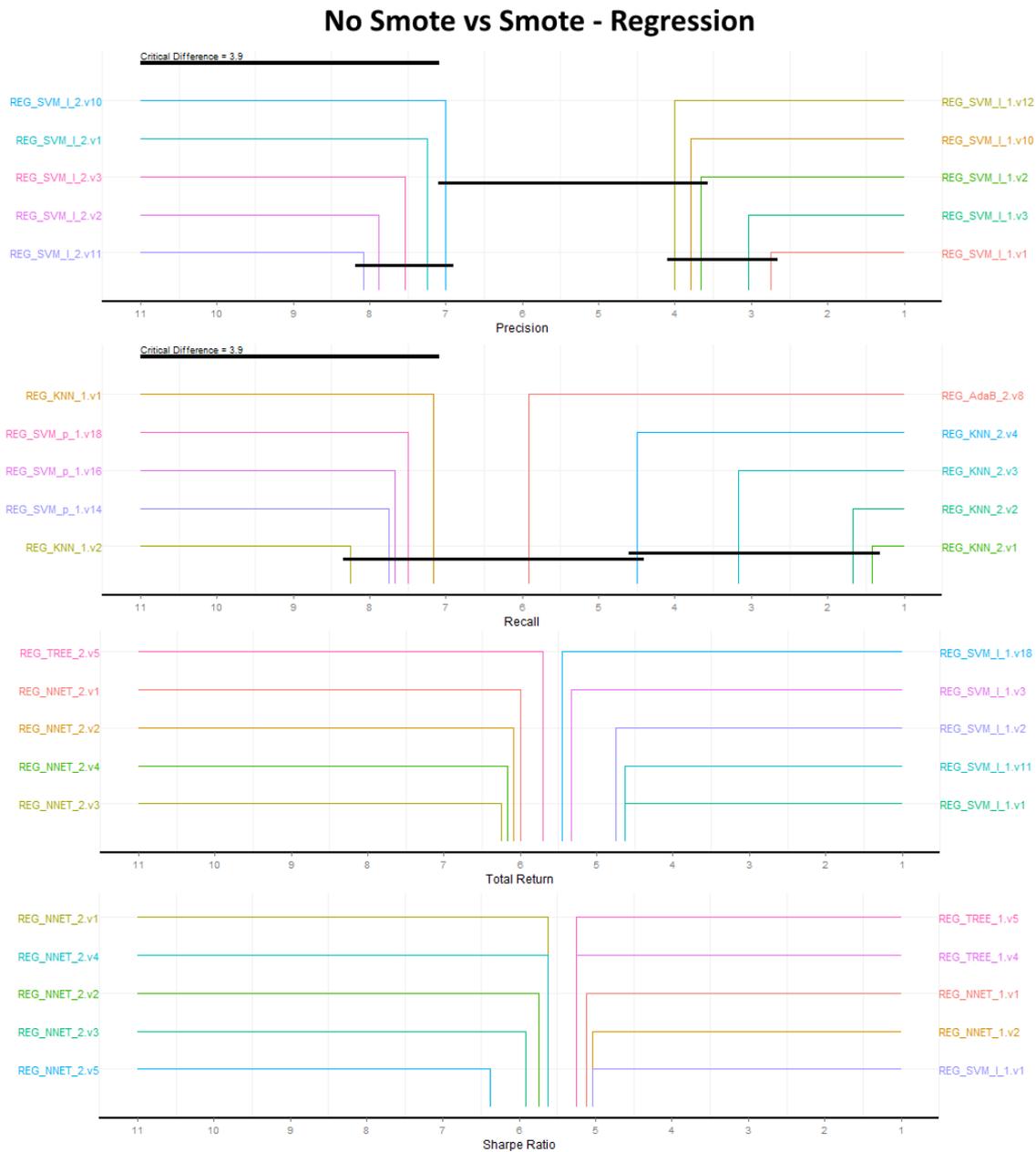


Figure 3.8: The top five average ranking model variants of both modelling alternatives (Regression with and without the usage of SMOTE) are forming a new set of variants, and their average rankings are re-calculated. Each model is thus given an average ranking (x axis). The presence of at least one black line implies that the Friedman's 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's statistical test.

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, since no money is either won or lost. Table 3.2 shows an example of such cost-benefit matrix that was obtained with the data from 1981-01-05 to 2000-10-13 of Apple. According to this particular evolution of the Apples’ shares, correctly forecasting a sell signals leads to an average profit of 4.9%, while correctly predicting a correct buy signals leads to a profit of 3.3%.

		Trues		
		s	h	b
Pred	s	4.9	-4.9	-8.2
	h	-2.4	0.00	-1.7
	b	-8.2	-3.3	3.3

Table 3.2: Example cost-benefit matrix for Apple shares.

Using this matrix and the probabilities of observation belonging to any of the classes that is produced by the used model, we can decide the final predicted class. Basically, for each possible class c , we multiply the probabilities vector by the line of the cost-benefit matrix corresponding to the class c . For each class, we obtain the expected reward if we forecast c . Therefore, the prediction will be the class with the highest expected reward. In Table 3.2, we can see how using this matrix may change the final predictions for 4 illustrative test cases. The first three columns are the obtained probabilities for each class by some classification model. The *Original Pred* is the class prediction of the model without using the cost-benefit matrix, which merely consists of selecting the class with highest probability. The *True Class* column is the correct signal while the *New Pred* is the class predicted if using the cost-benefit matrix together with the probabilities.

Prob of s	Prob of b	Prob of h	True Class	Original Pred	New Pred
0.21	0.42	0.37	s	b	h
0.09	0.21	0.70	b	h	h
0.09	0.44	0.47	h	h	b
0.17	0.44	0.39	h	b	h

Table 3.3: Illustration of the usage of the cost-benefit matrix into the original predictions of a classification model.

At the first line we have successfully prevented the most harmful mistake possible, the confusion between the sell and buy signals. Note that in this case, the original prediction was a buy signal, while the correct one was the sell. By using cost-benefit matrix, we could change the outcome to a hold signal, which despite being still wrong, it is definitely preferable. In the second observation, no changes have occurred in the model outcome. In

the following case, the new prediction actually forced a mistake while in the last one, it changed a wrong prediction into the correct one.

In order to test this hypothesis we will follow the same methodology of the previous cases. At the first, the top modelling variant will be tested against the ones in which the cost-benefit matrices were applied. The results regarding Precision and Recall are shown in Figure 3.9.

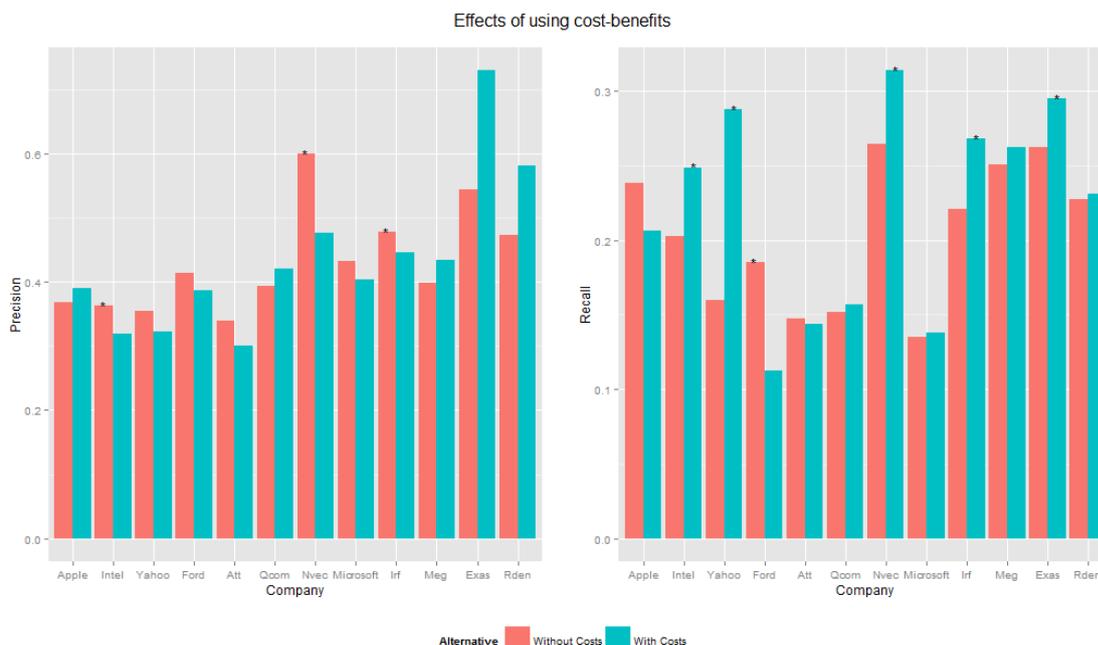


Figure 3.9: Best classification variant without costs against the best classification with costs for the macro versions of Precision and Recall metrics (asterisks denote that the respective variant is significantly better, according to a Wilcoxon test with $\alpha = 0.05$).

There is no clear conclusion from this figure. When it comes to the Recall, the use of these matrices led to better results, in which we see 5 significant positive differences against only one significant win of the simpler models. On the other hand, regarding Precision, we only see three statistical wins of the non-modified models. The reader may wonder why there are some cases with considerable differences without verifying the Wilcoxon statistical test, such as for the Exas company, but this is explained by the fact that the standard deviation of these results is quite high, thus implying that the respective scores vary too much. Nevertheless, it seems we are able to increase the recall of our models without severely compromising their precision by using cost-benefit matrices (at least when considering the best variant of all the models together).

Figure 3.10 shows the results of this experiment in terms of the financial metrics. The most evident observation is the fact that not a single significant win was achieved by either alternative. However, whether in terms of the Total Return or the Annualized Sharpe Ratio, the new models obtained more non significant wins. This is definitely a very interesting result, suggesting that the use of these matrices may be beneficial for the

classification models. Let us now check if these results hold across each individual type of model.

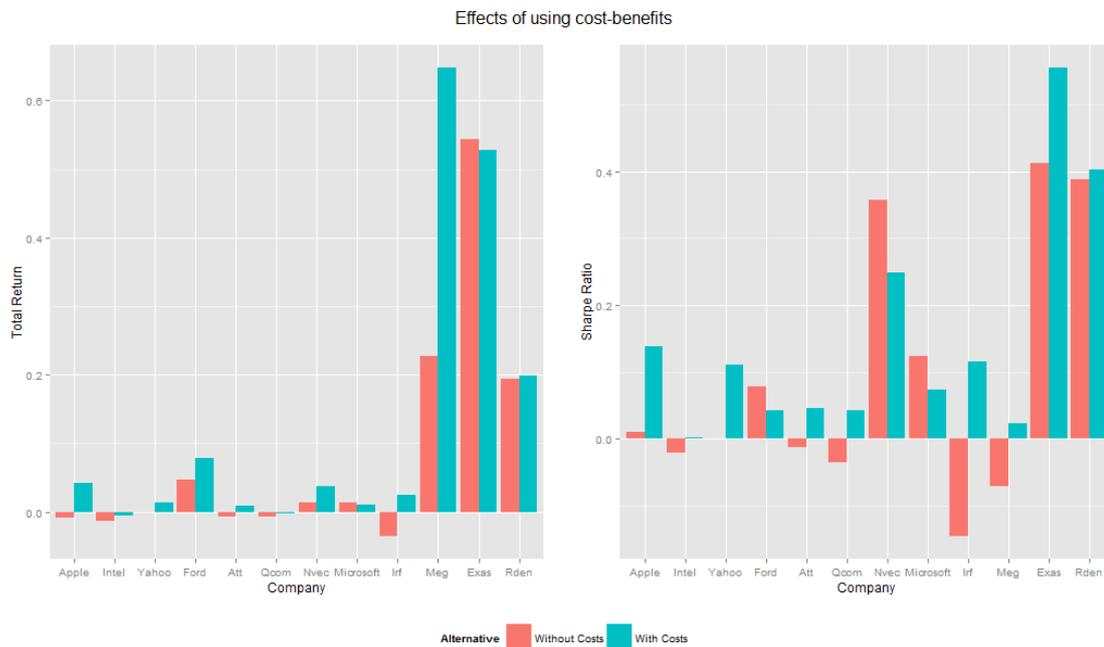


Figure 3.10: Best classification variant without costs against the best classification with costs for the Total Return and Annualised Sharpe Ratio metrics (asterisks denote that the respective variant is significantly better, according to a Wilcoxon test with $\alpha = 0.05$).

Figure 3.11 shows the comparison between the top modelling 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. While the top 3 types (NNET, TREE and SVM) are able to take advantage of the information on the matrices, the conclusions for the bottom 3 are not so clear.

Regarding the median performance of each type of model, we show the respective results in Appendix B.3. These results are quite interesting. Unlike the results for the top modelling variants per type of model, where we have observed very competitive results, in this new test there is some evident advantage of the standard models. It seems to be preferable to use the standard variants if the user is not willing to make an exhaustive search for the best parameters to model their financial problem.

Finally, we shown in Figure 3.12 the results concerning the average rankings of each modelling type, where he models names with “_2” are related to the ones created using cost-benefits. These results reflect what has been seen so far regarding the usage of cost-benefits: very competitive results across all the metrics. According to the results from this figure, there is no evidence supporting one alternative significantly outperforming the other, where the first five places are both occupied by models with and without cost-benefits.

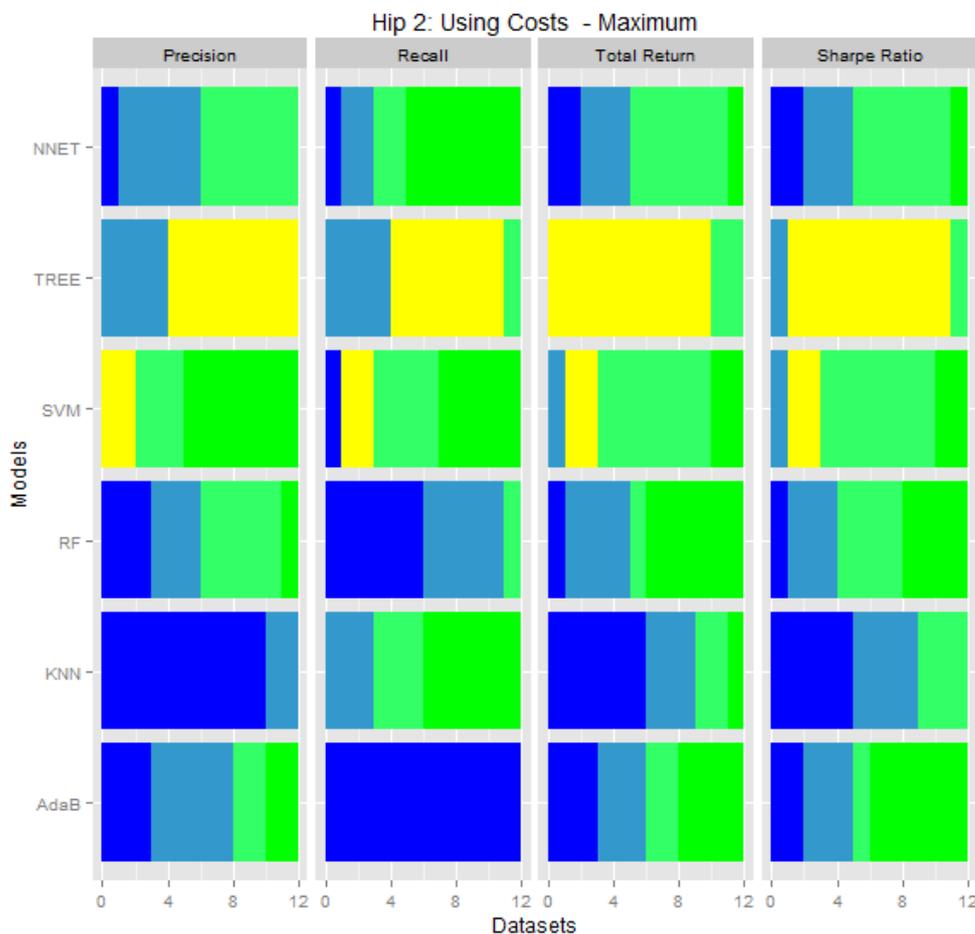


Figure 3.11: Segmented by type of model and by metric, a Wilcoxon test is performed between the best model variant of each modelling tool (Classification without costs vs Classification with costs). Since there are 12 data sets in each segment, then there are 12 results per segment. Each bar shows the results for each segment, where each one of the five colours 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 colour describes the number of times that type of win occurred.

3.3.4 Conclusions

We have empirically tested two hypothesis that we have put forward with the goal of trying to overcome some of the difficulties of the financial trading tasks.

The first hypothesis considered the usage of SMOTE as a tool to help the models to deal with unbalanced data led to very poor results in this financial context. Slightly better scores were obtained for the Regression models, although it still implied overall lower performances. Nonetheless, one or two type of models actually improved with SMOTE for a small set of companies. Still, the results of our experiments lead us to conclude that this does not seem an interesting procedure for this type of tasks and for the approaches and

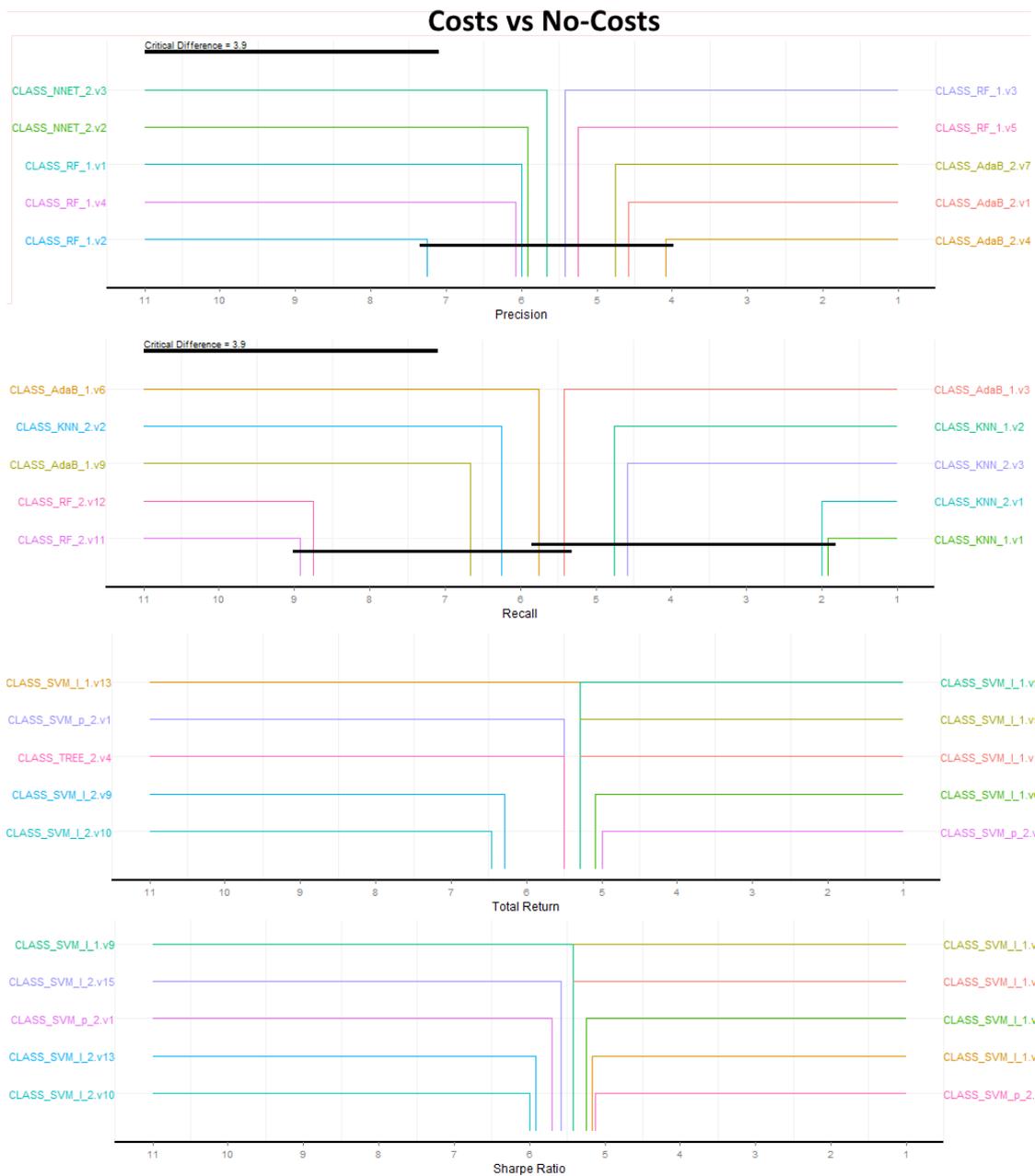


Figure 3.12: The top five average ranking model variants of both modelling alternatives (Classification with and without the usage of cost-benefit matrices) are forming a new set of variants, and their average rankings are re-calculated. Each model is thus given an average ranking (x axis). The presence of at least one black line implies that the Friedman’s 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’s statistical test.

models we have considered.

The second hypothesis involves the use of cost-benefit matrices with the Classification models, with the goal of giving them information on the implicit ordering among the classes, providing means for distinguishing the most distant classes (buy from sell). We have observed that, when considering the financial metrics, several classification variants

improved their performance. We have collected evidence suggesting that the addition of cost-benefit matrices will boost the overall performance of the classification modelling approaches.

It is important to remark that, particularly for the usage of cost-benefit matrices, we may expect some variability of the conclusions across different types of models. If we can afford to carry out an heavy search for the ideal parameter variants, then depending on the type of model, it may be preferable to apply costs or to use the standard models. This means that we can not say that we have observed a constant pattern of results for all the types of models. However, when using the median score across all parameter variants per type of model, it seems that standard models obtain better results. However, regarding the first hypothesis (the usage of SMOTE), most type of models are better without it, whether in terms of the maximum or median score per type of model.

Nevertheless, we have observed that for some particular setups (tasks and/or models) both hypothesis have improved the performance. In this context, in the remaining of our experiments we will not exclude the variants that use both strategies.

3.4 Comparison of Classification and Regression modelling approaches

This section presents the results of the experimental comparisons between the two general approaches to actionable forecasting in the context of trading decisions based on prediction 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, totalling $76 \times 3 = 228$ different classification variants. In terms of regression we have a slightly large set of 97 base models that where then tried with and without resampling, for a total of $97 \times 2 = 194$ variants. All these variants were compared on the data sets of the 12 companies using the methodology described in Section 3.2.

We will start by comparing the best model variant for each approach (classification and regression). As before, this initial study will be followed by a comparison of the average rankings of each model variant.

3.4.1 Wilcoxon test: Best per metric and per data set

For each company and for each metric, we have compared the top performer of both regression and classification modelling approaches using a Wilcoxon signed rank statistical test with a significance level of 0.05. 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. Similarly to the previous cases, the goal is to, task by task, compare the top modelling variant of each approach and observe any potential distinguishing behaviour.

Figure 3.13 shows the results of this comparison in terms of Precision and Recall of the sell and buy signals.

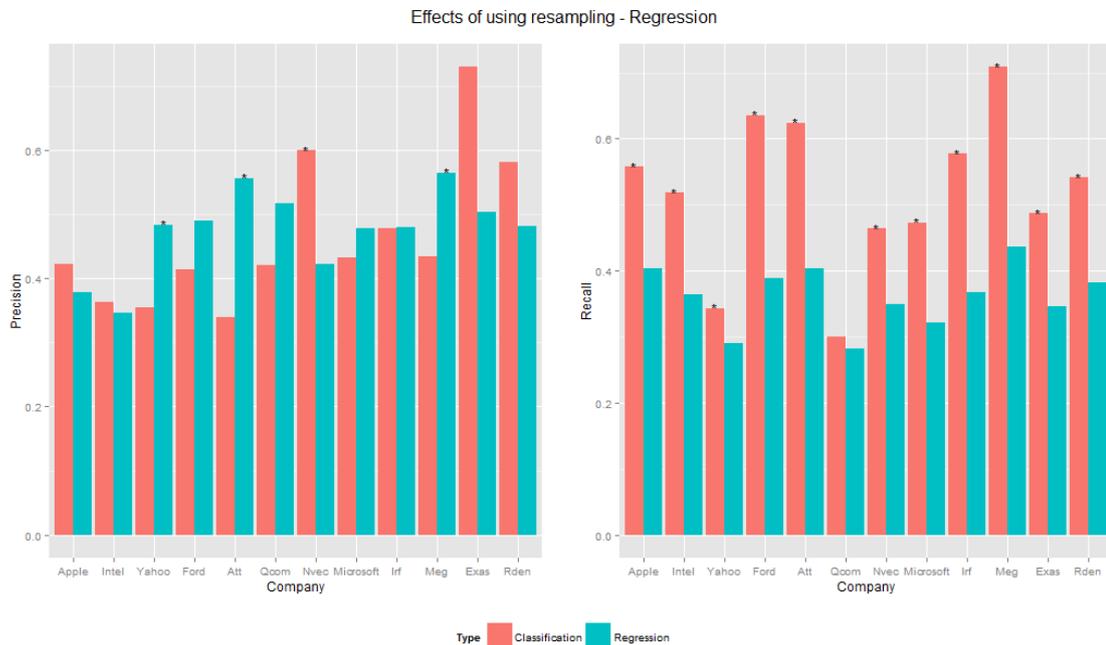


Figure 3.13: Best classification variant against the best regression one for the Precision and Recall metrics (asterisks denote that the respective variant is significantly better, according to a Wilcoxon test with $\alpha = 0.05$).

It is clear that the best performers in terms of Recall (graph on the right) are the models using a classification approach. This suggests that the first type may be able to detect more opportunities to make a profitable investment. However, if these higher values of Recall are not followed by high values of Precision, then these models may be risking too much and eventually lead to catastrophic losses. In terms of Precision (left graph), the difference is not so evident, although the regression modelling approach seems to be slightly better overall. Therefore, one may say that the latter may be more conservative. They capture a smaller percentage of all the profitable buy and sell signals, but when they do it, they seem to be correct more frequently than the classification approach, although we also see some companies with clear advantages of the latter.

Figure 3.14 shows the results of this comparison for the Total Return and Sharpe Ratio financial evaluation metrics. The results on the graphs of this figure 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 for 4 non significant 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 8 vs 1 non-significant wins. Overall, we have observed a very slight advantage of the best classification approach against the best regression variant.

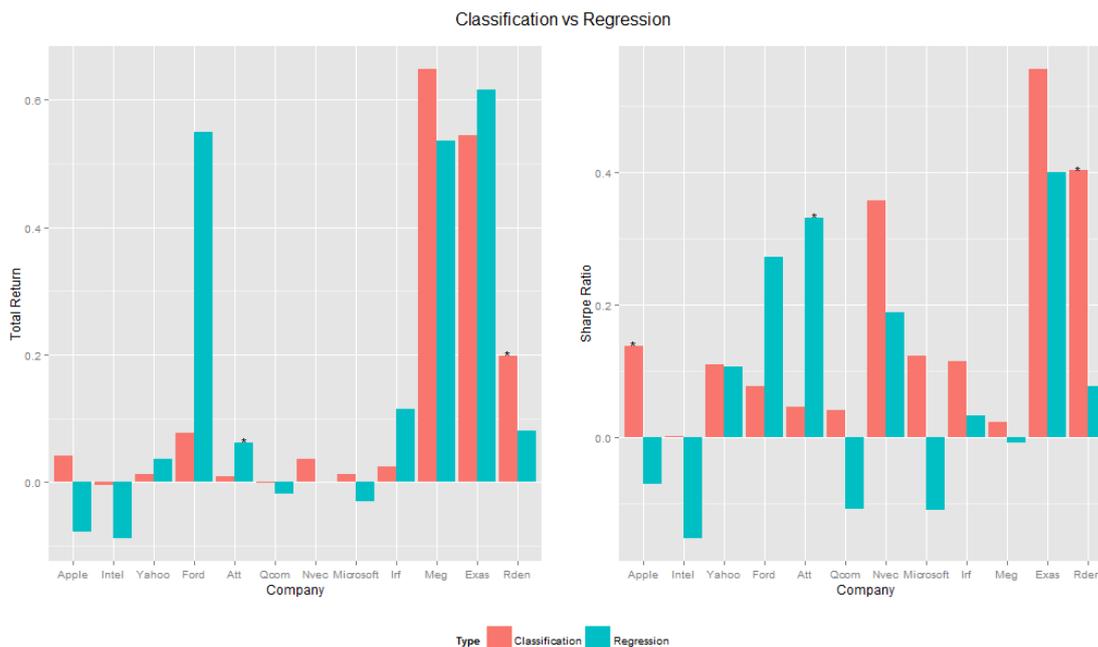


Figure 3.14: Best classification variant against the best regression one for the Total Return and Sharpe Ratio metrics (asterisks denote that the respective variant is significantly better, according to a Wilcoxon test with $\alpha = 0.05$).

From an economical perspective we have observed some contradictory results. For instance, for some companies (e.g. Meg) it was possible to achieve a very high level of Total Return (above 60% return), but the maximum Sharpe Ratio that was achieved 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 companies (e.g. Exas) for which both high values of Total Return and of Sharpe Ratio were reached, though we are not sure if these were achieved by the same model. 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 be misleading. This establishes the motivation for the second part of our experiments.

3.4.2 Post-hoc Nemenyi test: Average Ranks

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 approach across all the companies (top 5 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 significant differences among these average rankings of the top 5 variants of each approach. This way, if an approach 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 3.15 shows the results of this new comparison in terms of Precision and Recall. The results are consistent with the previous comparison. Across all the tasks simultaneously, the classification models are achieving on average better results when it comes to capture the highest amount of sell and buy signals while the regression models are making correct decisions more often when they forecast one of those two classes. However the difference in terms of Recall seems to be stronger (Nemenyi's test is statistically significant for some cases) than in terms of Precision. One should also note the following: the top 5 average-ranking models of either approach in the Recall metric is solely composed by models that were constructed using the re-sampling method SMOTE. This is theoretically expected, since the training set of these models was modified until all the classes were equally represented. Therefore, the model is more likely to predict the buy and sell signals, leading to higher values of recall, but also with the potential of making more mistakes (this behaviour was also observed when studying the results of applying SMOTE).

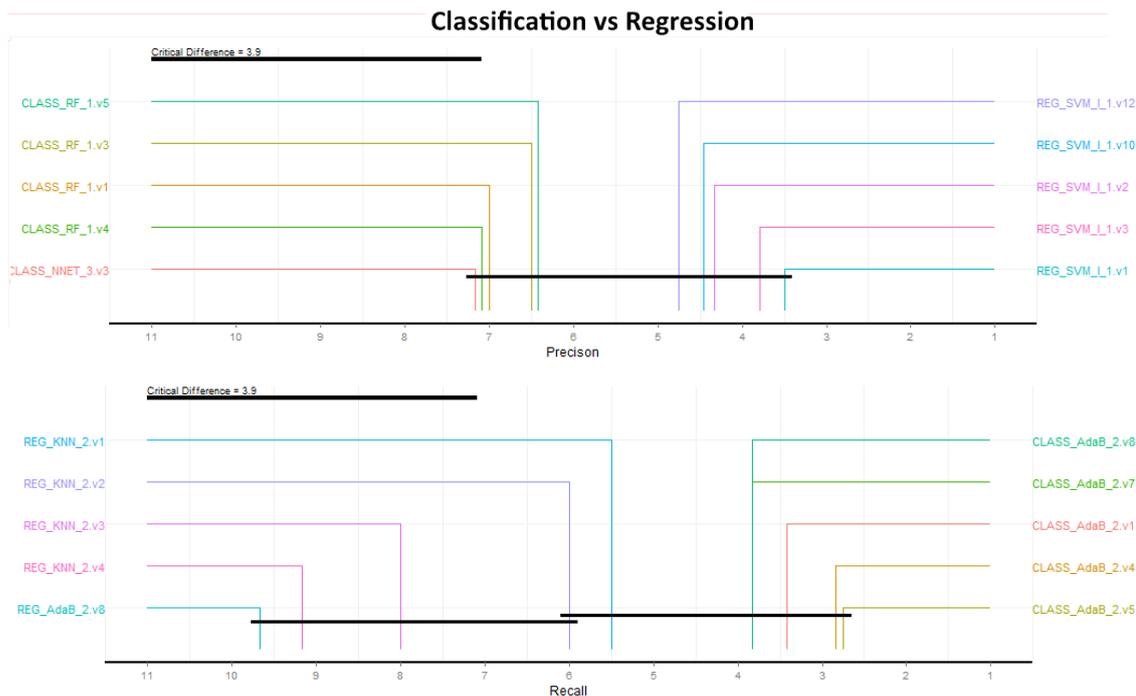


Figure 3.15: The top five average ranking model variants of both modelling approaches (Classification and Regression) are forming a new set of variants, and their average rankings are re-calculated. Each model is thus given an average ranking (x axis). The presence of at least one black line implies that the Friedman's 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's statistical test.

Figure 3.16 summarises the results in terms of Total Return and Sharpe Ratio. In both metrics, since we could not reject the Friedman's null hypothesis, the post-hoc Nemenyi's test was not performed. This means that we can not say with 95% confidence that there

is some difference in terms of Total Return or Sharpe Ratio between these modelling approaches. Nevertheless, there are some observations to remark. Regarding the first sub-image, i.e. the Total Return metric, 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. Moreover, not a single variant obtained using SMOTE appears in this top 5 for each approach, which means that we confirm that re-sampling does not seem to pay off for this class 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 can not say that any of the two approaches to actionable forecasting is better than the other in terms of Total Return.

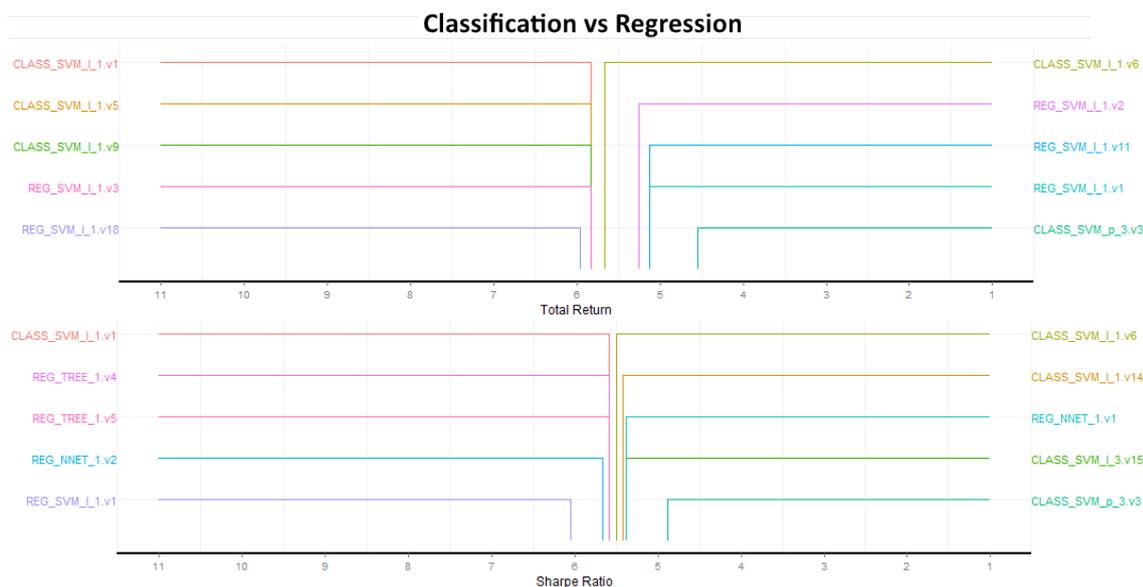


Figure 3.16: The top five average ranking model variants of both modelling approaches (Classification and Regression) are forming a new set of variants, and their average rankings are re-calculated. Each model is thus given an average ranking (x axis). The presence of at least one black line implies that the Friedman's 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's statistical test.

We analyse now the second sub-image, which shows the results of the same experiment in terms of Sharpe Ratio, i.e. 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 5 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, since 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 5 best results.

3.5 Conclusions

This chapter presented a 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 rules to choose the "correct" trading decision based on these predictions. The second approach tries to directly forecast the "correct" trading decision. This study is a specific instance of the more general problem of making decisions based on numerical forecasts detailed on the second chapter of the thesis, that we have named actionable forecasting. We have focused on financial trading decisions because this is a specific domain that requires specific trade-offs in terms of economic results.

Overall, the main conclusion of this study is that, we can not state that one approach performs significantly 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 is a 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. Given the large set of classification and regression models that were considered, as well as different approaches to the learning task, we claim that this conclusion is supported by significant experimental evidence.

The experiments carried out in this section have also allowed us to draw some other conclusions in terms of the applicability of re-sampling and cost-benefit matrices in the context of financial forecasting. Namely, we have observed that the application of re-sampling, although increasing the number of trading decisions made by the models, would typically bring additional financial risks that would make the models unattractive to traders. On the other hand, the use of cost-benefit matrices in an effort to maximise the utility of the predictions of the models, did bring some advantages to several modelling variants.

It is interesting to compare the conclusions of this specific study with the comparisons in a general setting described in the previous chapter. Were the conclusions from both studies consistent? The following are some of the most interesting observations from both studies:

- The usage of cost-benefit matrices was beneficial in both cases (generic and trading problems);
- While in the generic study the models with costs were better almost every time, in the trading problem the standard versions were more frequently better;
- In the generic study, the RandomForest type was unarguably the best modelling type, regardless of the modelling approach, while in the trading problem the SVM has occupied that place;

- The classification modelling approach had some considerable advantages in the generic study. In the trading problem, both approaches seem to be equally competitive.

The tasks considered in the trading problem present several distinct features from the ones in the generic problem, such as the temporal property as well as a strong unbalance of the response variable. These differences may explain some of the observed differences between the results of both chapters.

Chapter 4

Optimal Trading Signals

4.1 Introduction

In this thesis we have started by presenting an extensive comparative study between two possible approaches to Actionable Forecasting. This initial study aimed general tasks where decisions must be made based on numeric forecasts. We then focused on a particular instance of these applications: financial trading. This application has several particularities and it is sufficiently important to deserve this special treatment. In this chapter we continue our study of financial trading, but we now turn our attention to the key issue of how to evaluate the trading decisions.

Given a historic record of trading signals/decisions for the time-series of the prices of some company assets, it is not trivial to evaluate the quality of those signals. Merely looking at the total return obtained is not recommended since high returns may be obtained with large periods of time in which the investor would see their possessed assets with decreased value, i.e. the high return may be achieved at the cost of high variability of the returns (higher risk from the investor's perspective). In order to deal with this problem, there are other tools that can incorporate certain notions of risk, as the Sharpe Ratio, but typically, one can never look at a single metric and draw definitive conclusions. An analysis of several trading metrics must be conducted in order to build up some confidence regarding a trading system. There is another important drawback of existing metrics. The investors have different levels of risk aversion as well as different target returns, thus leading to the preference of different trading policies. To the best of our knowledge, there are no trading metrics that may be adapted to distinct trading policies, which means that each investor will have to perform a very subjective analysis from the scores of several standard trading metrics.

In this chapter, based on the work of [Torgo and Dhar \(2004\)](#), we propose a solution to this problem, that allows the establishment of which are the optimal trading actions given a certain target trading profile. These ideal actions can be used as a new form of evaluating trading systems by benchmarking their actions against this optimal performance.

We claim that the feedback provided by this benchmark is an interesting decision tool when evaluating trading performance. We present illustrative examples of the use of this benchmark for evaluating trading records.

4.1.1 Relationship with Activity Monitoring

The proposals described in this chapter are motivated by the concept of Activity Monitoring (Fawcett and Provost, 1999), more specifically by trying to see trading as an instance of these tasks. In activity monitoring data mining tasks one tries to find the right timings for issuing alarms for the so called positive activity that can be seen as target events one wants to signal timely. A standard example consists of detecting the presence of an intruder into someone's house. There is a period of positive activity, which is while the intruder is inside, and the goal is to set up an alarm the closest possible to the beginning of this positive activity. In this example, special attention must be paid to false alarms which are certainly unwanted. In this chapter, we show that financial trading can be described in a similar way with positive activity being the time windows where successful trades are possible, with a varying investor's definition of success. Given that high profits with minimal risks are the key issues for a successful trading record, we have used two metrics to capture these two properties as the means for defining a trader's definition of success. More specifically, we ask the trader to indicate the minimum wanted return for each individual trade (for instance to cover the transaction costs) and the maximum draw down she/he is willing to take (as a way to specify the maximum period of successive losses the trader is willing to accept).

In this context, a search for what would be the optimal timings regarding trading actions can be conducted. This search would be done maximising the total profit and making sure that the investor's trading policy is verified (in terms of the minimum return per trade as well as not surpassing the defined level of maximum draw down). It is also possible to find all the timings where a long/short position could be opened knowing that it would be possible to successfully close it (according to the same trading policy). Several uses may be given to this new perspective of trading. First of all, the performance of a real trading system may be compared against the performance of the optimal trading record (where these optimal signals would depend on a trading policy). Secondly, scoring functions may be used to measure the quality of the timings produced by the real trading system. This score could, for instance, also be used as the target variable of a financial forecasting system. In the thesis we focus on the first of these uses: traders benchmarking. We describe how to use this formalisation to obtain a benchmark score that we claim it is highly interpretable in the sense that it provides information on how distant is any trader from the optimal trading record given a certain target trading profile.

4.2 Definitions and Concepts

One of the main applications of our proposal is the creation of an adaptive benchmark that consists of the optimal trading positions according to the trader's preference criteria. This "adaptive benchmark" depends on the trading policy of the user, that will be defined according to three criteria:

- **Minimum wanted return** - The minimum profit percentage that the investor would like to obtain per trade;
- **Maximum Draw Down Allowed** - It measures the largest peak-to-trough decline in the value of a portfolio (before a new peak is achieved). Suppose a long position is opened with a price p . At every point in time we consider the maximum value p_{max} that was obtained between opening the position and the present time. If at any moment, the present value of the asset is lower than the maximum draw down allowed times p_{max} , then the maximum draw down allowed level is breached. In other words, it describes how much devaluation we are willing to take before closing the position in order to prevent bigger losses. This criteria describes somehow the risk aversion of the client;
- **Transaction Cost** - A percentage cost that is assigned to a trade that closes a long or short position. For instance, consider the opening of a long position with a price equal to 90 and the closing with a price of 100. With a transaction cost of 1%, the true return would be $((1 - 0.01) \times 100) - 90 = 99 - 90 = 9$.

In order to illustrate the influence of the minimum wanted return and the maximum draw down allowed, we show in Figure 4.1 an example that would lead to a forced sell. The y axis represents the values of the assets, with d_{alpha} being the opening price of a long position. Since the growth of the price was not enough to surpass the target minimum return (r_g), we could not close the long position. When the asset's price reached the d_{max} value, a fall was observed. This fall was high enough to surpass the maximum draw down allowed (mx_{DD}), forcing the close of the long position in order to prevent more losses.

Different values of minimum wanted return and maximum draw down allowed could have made this unwanted situation into a more desirable one. Decreasing the value of the first criteria (target return) could have let the position to be closed before the fall of the price or, on the other hand, different values for the maximum draw down allowed could have lead to fewer losses (by having a lower mx_{DD} which would force the sell sooner) or allowed the asset's value to increase once again (with a higher mx_{DD} , since the investor would not sell at that timing). We believe that with these variables we can make a very realistic approximation of the many possible investor's trading policies. In this example, no transaction costs were used.

In order to apply the methodology being described in this chapter, it is important to know the timings of all the trading actions settled by a trading system. In our study we

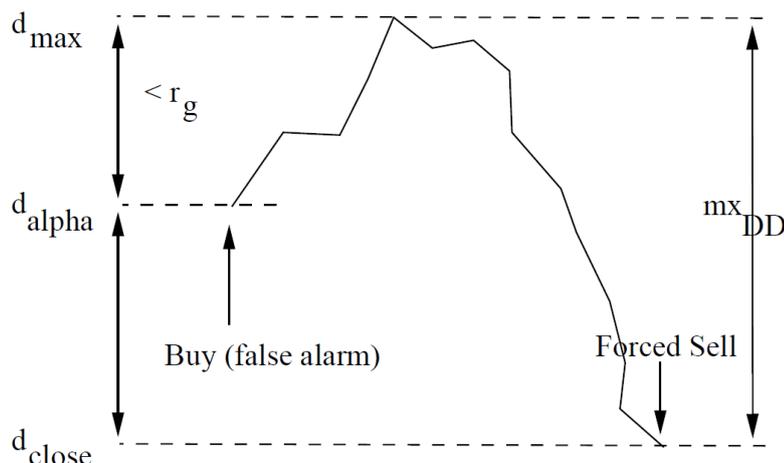


Figure 4.1: A long position opened by a false alarm. This image was taken out from [Torgo and Dhar \(2004\)](#)

consider two types of trading positions: long and short. The first is opened by issuing a Buy order to obtain assets/securities and it is closed when a Sell order is issued. The goal is to sell those assets/securities at some time in the future at a higher value, thus leading to a profit. On the other hand, a short position is opened by issuing a Sell order (even though the investor may not possess any asset, which is possible through a borrowing system) and it is closed when the investor buys the securities. In other words, the investor sells a security at the price of the current time, with a promise to buy the respective assets at some point in the future. If the security's value decreases during that time, then a profit is obtained. The key for successful trading is to open and close these positions at the right timings.

4.3 Optimal Positions

We will describe the procedure to construct the optimal benchmark according to a certain trading policy. For a certain evolution of the prices of some assets, the goal is to construct the set of long and short positions (the moments of opening and closing) that would be ideal for a certain trading policy. We will describe the procedure for obtaining these ideal timings separately for each type of position. Please note that this concept of optimal positions can only be implemented a-posteriori, i.e. the optimal positions can only be calculated for a past period, after we know what happened to the market prices, which means that there is no forecasting involved in this benchmarking procedure.

4.3.1 Long Positions

In order to obtain the optimal timings for executing long positions, one must look at the periods where there are no draw downs larger that the one specified in the trading policy,

mx_{dd} , and search for the opening and closing timings within that period that lead to the largest profit, as long as that profit is higher than the target return r_g . Algorithm 1, that was firstly conceptualised in the work of Torgo and Dhar (2004), returns a vector with the optimal trading record regarding long positions. For every unit of time, the returned vector will have one of three following values: {"hold","open","close"}. The first means that no trading actions should have occurred at that timing, i.e. no positions should be opened nor any active position should be closed. The second means that a long position should have been opened on that timing while the latter means that the current long position should be closed.

This algorithm is very efficient with a computational complexity of the order of $O(n)$. We now briefly describe the intuition behind the algorithm. Variables "mn" and "idx.mn" are related to the minimum price observed during a certain period. While the first is the value itself, the latter is the timing where the minimum has occurred. Similarly, "mx" and "idx.mx" are related to the maximum observed in the same period, where the first is the value and the second is the timing where that price has occurred. As detailed before, we need to look for periods between draw downs larger than mx_{DD} , where a return larger than r_g occurs. This search consists in finding the maximum and minimum prices within those periods. The variables just described will contain the information regarding those extreme prices during each potential period to trade. From line 3 to 9, the algorithm tests if the current timing corresponds to a new maximum. If so, we immediately update the variables that store the information regarding the maximum. It is also tested whether the difference between the new maximum and minimum value stored in "mn" is high enough to cover that target return plus the transaction cost. In that case, we update the variable "gain" to have the value "True". At this moment, opening a long position at the timing "idx.mn" and closing at the timing "idx.mx" would correspond to a successful trade. However, it may not be an optimal one, since the market may still go up in the following timings. While we wait to check if this happens, we say that there is a *standby long position*.

Lines 10 to 15 test if the current timings corresponds to a new minimum. If they are, then we are now sure that any eventual *standby long position* (produced in the previous lines) may now be considered as an optimal one. This happens because if a new minimum occurs, it would not make sense to have a long position starting before the timing of the minimum and ending after it. Therefore, if the variable "gain" is equal to *True*, we add a new long position to our final vector of signals, using the timings of minimum and highest prices observed before. Moreover, we update our minimum and maximum temporary variables to the current timing, somehow "restarting" the process of finding the next optimal long position.

Finally, the maximum draw down component is taken care between lines 17 and 21. The first condition tests if the maximum draw down requirement was breached. If that is the case, then we can no longer wait for the market to go up. Therefore, we consider any eventual *standby long position* as an optimal one. Similarly, when a new minimum is

encountered, if the maximum draw down is compromised, then we update the maximum and minimum temporary variables to the current timing, because a long position starting now would always be better than one starting before this timing and still being currently active.

When the For loop ends, implying that we have reached the end of the test time-series, we still test if there was any *standby long position* active. In that case we add the opening time to the final vector, but not any closing timing since it could be possible to obtain an even more profitable trade in the upcoming period.

Algorithm 1 Signals corresponding to Ideal Long positions.

```

1: function SIGNALSIDEALLONG( $x, r_g, mx_{dd}$ )
   Input :  $x$ , a sequence of prices of past sessions
            $r_g$ , the minimal return required for all positions
            $mx_{dd}$ , the maximal allowed draw down
            $t_c$ , the transaction cost
   Output :  $sig$ , a vector of signals, one for each session

2:    $n \leftarrow \text{length\_of}(x)$ ;  $mn \leftarrow mx \leftarrow x[1]$  ;  $idx.mn \leftarrow idx.mx \leftarrow 1$  ;  $gain \leftarrow \text{False}$ 
3:   for  $i = 1$  to  $n$  do
4:      $sig[i] \leftarrow \text{'hold'}$ 
5:     if  $x[i] > mx$  then
6:        $mx \leftarrow x[i]$  ;  $idx.mx \leftarrow i$ 
7:       if  $(mx - mn)/mn \geq r_g + t_c$  then
8:          $gain \leftarrow \text{True}$ 
9:       end if
10:    else if  $x[i] < mn$  then
11:      if  $gain$  then
12:         $sig[idx.mn] \leftarrow \text{'open'}$  ;  $sig[idx.mx] \leftarrow \text{'close'}$  ;  $gain \leftarrow \text{False}$ 
13:      end if
14:       $mn \leftarrow mx \leftarrow x[i]$ ;  $idx.mn \leftarrow idx.mx \leftarrow i$ 
15:    end if
16:    if  $(mx - x[i])/mx > mx_{dd}$  then
17:      if  $gain$  then
18:         $sig[idx.mn] \leftarrow \text{'open'}$  ;  $sig[idx.mx] \leftarrow \text{'close'}$  ;  $gain \leftarrow \text{False}$ 
19:      end if
20:       $mn \leftarrow mx \leftarrow x[i]$ ;  $idx.mn \leftarrow idx.mx \leftarrow i$ 
21:    end if
22:  end for
23:  if  $gain$  then
24:     $sig[idx.mn] \leftarrow \text{'open'}$ 
25:  end if
26:  Return  $sig$ 
27: end function

```

The outcome of Algorithm 1 allows us to obtain the optimal periods of time for holding a long position given a selected trading profile. Figure 4.2 shows these optimal long positions

for the prices of S&P 500 from May of 2012 till April of the following year. The optimal periods where we should be “long” are marked in blue, where the beginning and ending of each blue rectangle correspond to the optimal opening and closing timings. These were calculated using Algorithm 1 given two different trading policies: (i) $r_g = 2\%$ and $mx_{dd} = 1\%$; (ii) $r_g = 5\%$ and $mx_{dd} = 3\%$. As we can see the optimal trading record is different for each case. The first allows more trades to be executed since with a reduced maximum draw down allowed, it will be harder to keep a position opened for a long time, thus leading to more and shorter trades. The second case presents long positions opened for a very long time since the investor was willing to take more risks. However, with a higher target return, there will be several chances missed in which the investor could obtain smaller yet positive returns.



Figure 4.2: Optimal long positions given two different trading policies of S&P 500 between July of 2012 and April of 2013.

4.3.2 Short Positions

The construction of the optimal short positions is quite similar to the long ones. The main differences are the way we obtain a return as well as the way we test if the maximum draw down allowed was breached. In terms of return, the investor is now interested in negative variations of the prices, while the maximum draw down allowed is now breached due to the potential increase of the prices. Algorithm 2 has the required steps to obtain these ideal short positions.

Algorithm 2 Signals corresponding to Ideal Short positions.

```

1: function SIGNALSIDEALSHORT( $x, r_g, mx_{dd}$ )
   Input :  $x$ , a sequence of prices of past sessions
            $r_g$ , the minimal return required for all positions
            $mx_{dd}$ , the maximal allowed draw down
            $t_c$ , the transaction cost
   Output :  $sig$ , a vector of signals, one for each session

2:    $n \leftarrow \text{length\_of}(x)$ ;  $mn \leftarrow mx \leftarrow x[1]$  ;  $idx.mn \leftarrow idx.mx \leftarrow 1$  ;  $gain \leftarrow \text{False}$ 
3:   for  $i = 1$  to  $n$  do
4:      $sig[i] \leftarrow \text{'hold'}$ 
5:     if  $x[i] < mn$  then
6:        $mn \leftarrow x[i]$  ;  $idx.mn \leftarrow i$ 
7:       if  $(mx - mn)/mx \geq r_g - t_c$  then
8:          $gain \leftarrow \text{True}$ 
9:       end if
10:    else if  $x[i] > mx$  then
11:      if  $gain$  then
12:         $sig[idx.mx] \leftarrow \text{'open'}$  ;  $sig[idx.mn] \leftarrow \text{'close'}$  ;  $gain \leftarrow \text{False}$ 
13:      end if
14:       $mn \leftarrow mx \leftarrow x[i]$ ;  $idx.mn \leftarrow idx.mx \leftarrow i$ 
15:    end if
16:    if  $(x[i] - mn)/mn > mx_{dd}$  then
17:      if  $gain$  then
18:         $sig[idx.mx] \leftarrow \text{'open'}$  ;  $sig[idx.mn] \leftarrow \text{'close'}$  ;  $gain \leftarrow \text{False}$ 
19:      end if
20:       $mn \leftarrow mx \leftarrow x[i]$ ;  $idx.mn \leftarrow idx.mx \leftarrow i$ 
21:    end if
22:  end for
23:  if  $gain$  then
24:     $sig[idx.mx] \leftarrow \text{'open'}$ 
25:  end if
26:  Return  $sig$ 
27: end function

```

Similarly to the long positions, we present for the same time series and trading policies, the outcome of Algorithm 2. The optimal periods where we should be “short” are now marked in orange, where the beginning and ending of each orange rectangle corresponds to

the optimal opening and closing timings. Given the same two trading policies used before: (i) $r_g = 2\%$ and $max_{dd} = 1\%$; (ii) $r_g = 5\%$ and $max_{dd} = 3\%$, we have obtained in Figure 4.3 the optimal short positions. Since there is a clear positive trend in the evolution of the prices, there are less chances for successful short positions, thus explaining the reduced number of trades compared to the long ones. Once again, and just as expected, using the first trading policy leads to more trades while the second produces less trades but each with potentially much higher returns than the first ones.

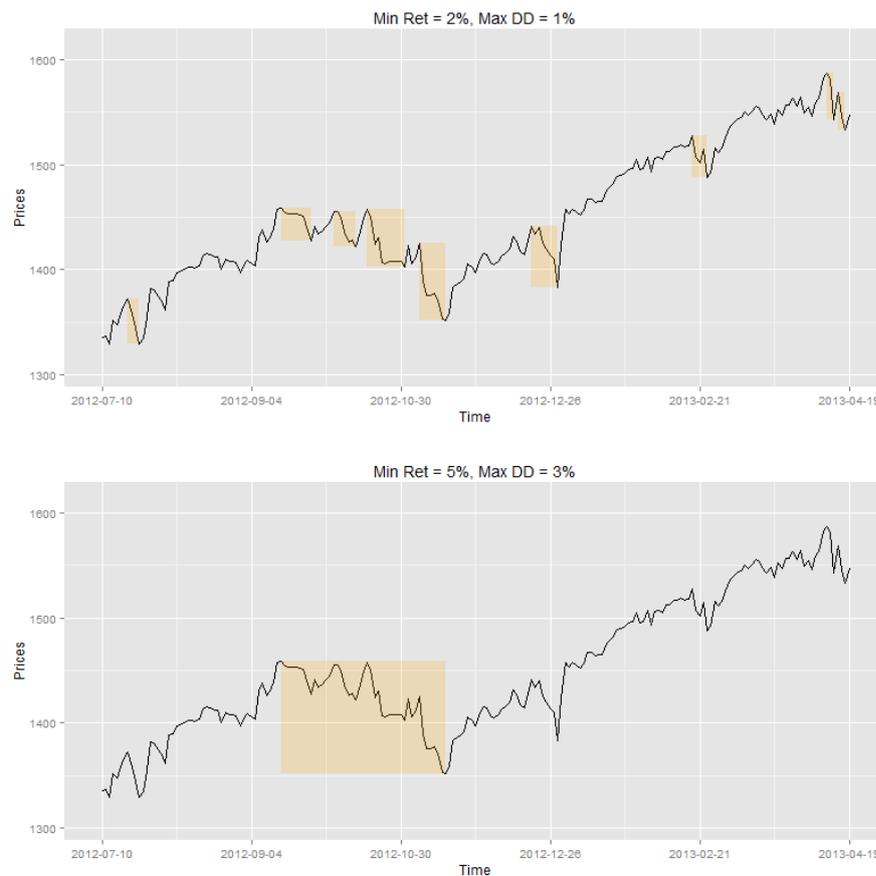


Figure 4.3: Optimal short positions given two different trading policies of S&P 500 between July of 2012 and April of 2013.

Using the previous two algorithms we can combine their outcomes to produce the optimal trading record for this example test period. One may also wonder if the quantity of the produced long and short positions is not low, since nowadays trend is to perform high frequency trades. However it all depends on the trading policy, where lower values of minimum wanted return and maximum draw down allowed may dramatically increase the total number of positions opened for the same time interval. Moreover, we are using days as our time unit. By changing to hours or even smaller unit of time, the number of positions obtained will inevitable increase.

4.4 Positive Activity for Trading

We have seen how to construct a set of optimal signals according to a trading policy. There is some other valuable information that can be provided to an investor based on such a trading policy. On top of providing the ideal timings for opening and closing the positions, we may also determine all time stamps on which if a position is open it may lead to a successful position, i.e. opening at those times a position it is possible to obtain the intended return without going through the maximum draw down, provided the position is closed in the write time. The periods where this is possible correspond to our notion of *positive activity*. Note that those calculated timings only say that there would be at least one corresponding closing timing that would lead to a successful position. In other words, it says whether opening a position at a certain timing was the right call or not, but the decision of closing that position still needs to be conducted.

The information provided by this positive activity may serve several purposes. One may check the performance of a trading system by analysing, for instance, the percentage of times that the trading system has opened a position in a positive activity period. One may also assign a score function to every time stamp (for instance depending on whether they belong to the positive activity period or on the potential maximum return that could be obtained by opening a position on each timing). With these scoring functions, the investor may evaluate the timings of the positions produced by a trading system. These scores could also be used outside of the context of traders benchmarking. For instance, the score could be the target variable of a machine learning model, where correctly forecasting these values would help to create a successful trading system, that would actually be adapted to the investor's trading policy. In this thesis, however, we do not address these other potential uses of our proposed method based on defining trading as activity monitoring.

Each position has two different events that a trading system should detect. One is the time where we should open the position, and the other is the time where we should close it (where the closing process naturally depends on the opening timing of that position). Each of these two times should be issued as close to the optimal timings as possible. In this context, each of them has an associated positive activity period, which can be regarded as the period where opening or closing a position is successful according to a trading policy (even if it is not ideal). We claim that it is much more important to focus on the timings for opening positions rather than closing for two reasons: (i) the positive activity for closing a position naturally depends on the opening timing, and therefore, if the opening time does not belong to a period of positive activity, irrespectively of the closing time, the investor will never be able to successfully close, and thus there will be no positive activity for the closing position; (ii) several trading systems just focus on properly opening long positions, where the closing moment automatically occurs when a certain price is verified (through market orders with target profit and with stop-loss orders). In this sense, the quality of a trading system is highly dependent on their opening timings. Therefore, we will give more

emphasis to the positive activity regarding the proper timings to open a position.

We will start by analysing the positive activity for opening long positions (short positions will be analogous). A period of positive activity is a period where a buy order (that corresponds to the opening of a long position) would be useful in the sense that it would be possible for the target return in the future to be attainable without going through the maximum draw down that was set.

Given a trading profile characterised by a target return of r_g and a maximum draw down of mx_{dd} we can formally describe the notion of positive activity for opening a long position as follows. We can define it as a period $P = \langle d_x, \dots, d_y \rangle$ (set of timings) that verifies the following properties: $\forall d_i \in P, \exists m : m > i \wedge \frac{d_m - d_i}{d_i} \geq r_g \wedge Max_{DD}(\langle d_i, \dots, d_m \rangle) < mx_{dd}$. Less formally, a period of positive activity is a period where opening a position is able to capitalise a return of at least r_g , before going through a draw down of mx_{dd} . Regarding positive activity for short positions the definition is similar, with the second condition replaced by $\frac{d_i - d_m}{d_i} \leq -r_g$.

Figure 4.4 shows the positive activity periods for opening long positions based on two different trading policies. This example test period as the well the trading policies are the same as in Figure 4.2. The periods of positive activity were obtained using the definition provided above and are shown in blue. For instance, for the first policy if a buy order is issued within any period marked in blue, then it is possible to achieve a gain of 2% before going through a draw down of 1%.

The concepts used so far are intimately connected with the theory of activity monitoring. However, when applying this theory to trading tasks, there are some differences that must be considered (namely from the framework presented in [Fawcett and Provost \(1999\)](#)). While the goal of activity monitoring is to detect an event as fast as possible (remember the intruder's example), here this rule may not longer be valid. The starting moment of a positive activity period for opening a position may not be the one that leads to the largest profit, even though it would be successful according to the used trading policy. Moreover, unlike the generic activity monitoring tasks, the positive activity periods are no longer necessarily contiguous in time. There may be periods where opening a position would not viable according to the trading policy for a very small margin, while in the days after and before it already would be. In Figure 4.4, we can see both these differences. Notice how the beginning of the first period of positive activity occurs while the market is still decreasing. The optimal position would be at the lowest peak and that would be the timing we would like to forecast instead of the first moment of positive activity period. The discontinuities are also clear in this figure.

Obtaining the positive activity for the short positions would be analogous in every way where positive returns occur when the asset's value decreases.

As mentioned before, there are some side applications for this concept of positive activity. The one with the highest potential is the creation of scoring functions. A score can be calculated at every point in time according to whether they correspond to positive activity

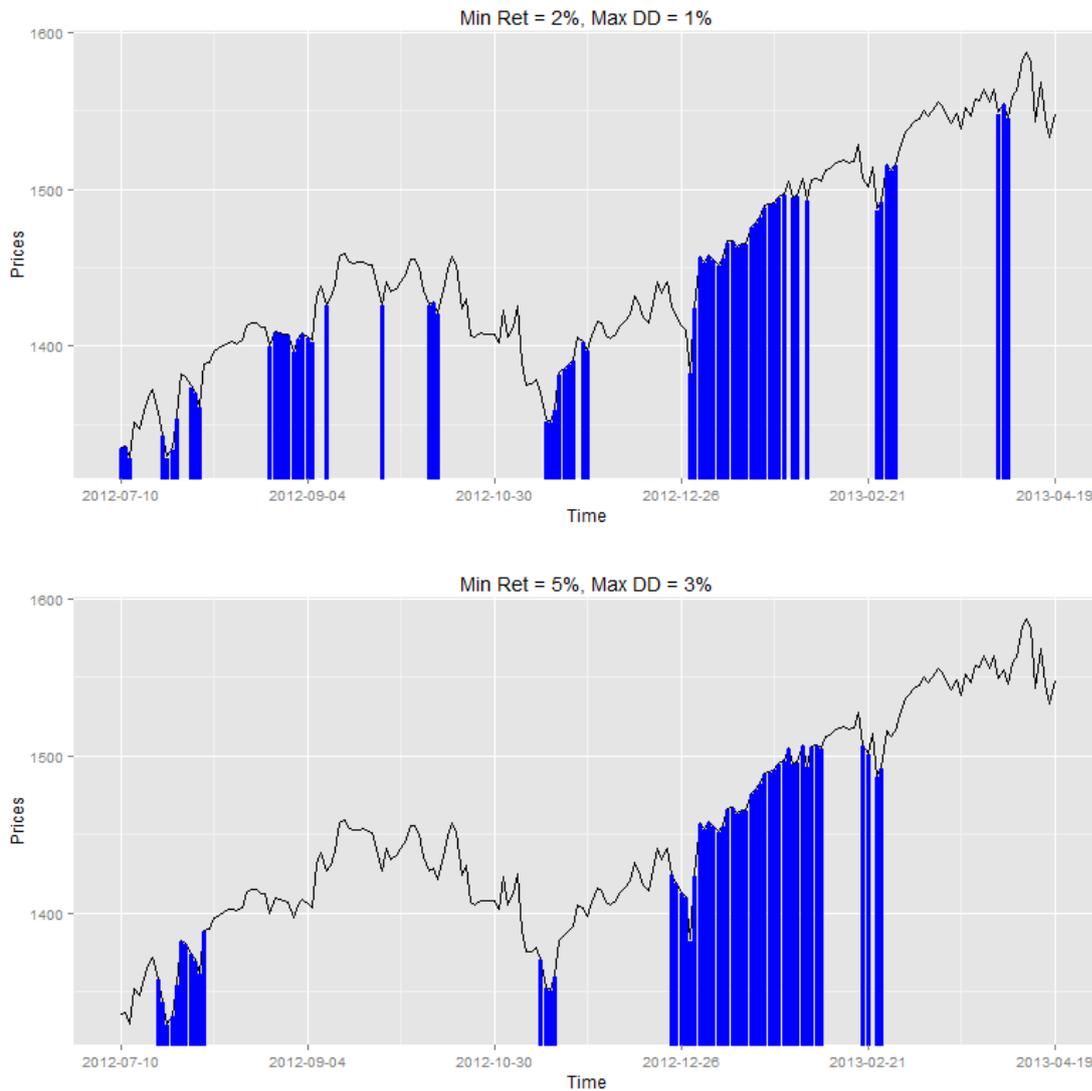


Figure 4.4: Positive Activity of S&P 500 between July of 2012 and April of 2013 for two different trading policies.

or not, and if they do, how close would that timing be to the optimal time for the signal. Using that score, one may evaluate a trading system by seeing the scores associated to the timings that the system has opened a position. This would be a whole new perspective of evaluating a trading system. Not only would it be quite adapted to a trading policy, but also it would focus on the proper timings rather than merely at the distribution of the returns.

We list some scoring functions that may be considered for long positions, starting by a trivial example:.

- **Function a)** - Positive Activity,

$$f: T_l \rightarrow \mathcal{S} = \{0, 1\}$$

$$t \mapsto \begin{cases} 1, & t \in P_l \\ 0, & t \notin P_l \end{cases}.$$

with T_l representing the set of timings that the system has opened a long position and P_l the activity period of a long position. This score function says whether a long position was opened in a time of positive activity or not (and therefore if it would be possible to obtain a profit or if a forced sell would inevitably occur).

- **Function b)** - Temporal distance to the closest optimal signal,

$$f: T_l \rightarrow \mathcal{S} = \{0, 1\}$$

$$t \mapsto \begin{cases} 1 - d(t, t_{opt}), & t \in P_l \\ 0, & t \notin P_l \end{cases}.$$

where $d(t, t_{opt})$ would be a normalised distance function between the timing t and the closest optimal timing for opening a position. The score of each point in the positive activity period would be decreased as higher the distance to the closest optimal timing for opening a long position, or zero if a forced sell would have to inevitably occur.

- **Function c)** - Ratio between the potential return of a time t with the closest optimal signal,

$$f: T_l \rightarrow \mathcal{S} = \{0, 1\}$$

$$t \mapsto \begin{cases} \frac{r(t)}{r(t_{opt})}, & t \in P_l \\ 0, & t \notin P_l \end{cases}.$$

where r is a function that measures the potential return that may be obtained by opening a long position at the time t . This potential return is measured by considering the closing position that would lead to the largest return possible without compromising the trading policy used. The score of each point within the positive activity period would increase as higher as the potential return of that time compared to the return associated to closest optimal position.

- **Function d)** - Ratio between the potential return of a time t with the closest optimal signal also considering the losses,

$$f: T_l \rightarrow \mathcal{S} = \{0, 1\}$$

$$t \mapsto \begin{cases} \frac{r(t)}{r(t_{opt})}, & t \in P_l \\ -r_{loss}(t) + r_g, & t \notin P_l \end{cases}.$$

where r_g is the target return and the function $r_{loss}(t)$ calculates the maximum value of the security's prices between the timing t and the moment of forced sell. In other words, the score assigned to timings that do not belong to a positive activity period can be interpreted as how much more positive variation of the prices was needed in order for the timing t to be successful according to the used trading policy.

Any of these functions can be used to assign a score to the timings for opening long positions issued by any trading system. Analogous functions may be created for short positions. Using the set of scores obtained by a trading system along time, one may conduct a study regarding the distribution of the scores of this trading system. Depending on the chosen function, one may carry out statistical tests to check if that distribution of the scores of the trading system has some desired properties. For instance, suppose we are considering the trivial function described above (0 or 1 depending if the timing belongs to the positive activity period). In this case, we could be interested in carrying out a statistical test to check if the average would be significantly higher than 0.5.

We claim that the proposed methodology based on looking at trading as activity monitoring may open several interesting new avenues regarding ways of evaluating a trading system performance, particularly by allowing to check this performance against some user preference bias (i.e. the user trading policy). Another potential interest of these scoring functions is that they may enable new ways of constructing trading systems. Given an historical record of some security's prices, we may calculate the score for opening a position on each point in time according to the trading policy of the investor. Then, those scores could be used as the target variable of a machine learning model. The obtained model could forecast the score for some future time stamp and based on this score trading decisions can be made.

4.5 Optimal Signals as a Trading Benchmark

We have described how to obtain the optimal signals for some past period of time and given the trader's preference biases. The motivation for finding these signals would be to create a benchmark that an investor could use to compare several alternative trading systems and choose the most adequate according to his trading policy. In this section we will explain how to use the optimal signals as a benchmark, list some of the existing

evaluation metrics, and describe a practical application of the proposed benchmark to the signals of a real trading system.

4.5.1 How to use the Optimal Benchmark

We now describe how the optimal trading signals can be used to evaluate a trading system. We assume that this trading system was applied to some (past) testing period. Using some specific target trading policy (defined by the target return and maximum allowed draw down) we calculate the optimal trading systems using the algorithms described before. Having these optimal signals and the signals produced by a concrete trading system, we propose the following methodology to evaluate this trading system according to the optimal portfolio:

- For any trading evaluation metric (e.g. total return, profit factor, sharpe ratio, etc.), we calculate two values: (i) the value obtained by the signals of the system being evaluated; and (ii) the value obtained by the optimal trading signals. Using these two values we calculate the ratio between them. We claim that these ratios are highly interpretable for a trader because they reflect how far is the system being evaluated to the ideal signals according to the trader target policy. For instance, assume we are considering some risk measure, such as the well known Sharpe Ratio. Suppose the ratio of the Sharpe score of the trading system over the Sharpe obtained by the optimal signals is 0.1. This means that the trading system has 90% more risk exposure than the score that would be possible to obtain on the testing period, assuming the trader targets in terms of return and maximum draw down.

We claim that this proposed evaluation methodology is quite informative, highly interpretable and easy to implement. Our proposed benchmark may also open doors for new metrics. To the best of our knowledge, there are very little metrics that try to measure in some way, how distant is a concrete trading record from what would be ideal performance by an investor. Usually, it is the opposite way, where the trading records are evaluated according to how better they are against a very simple benchmark.

4.5.2 Standard Metrics for Evaluating Trading Systems

We have suggested the use of several standard trading evaluation metrics to obtain ratios over the score of the optimal trading actions. In this section we describe some financial metrics that could be used in this context. We will divide our metrics into two groups: (i) metrics that only look at the final return upon closing a position; and (ii) metrics that consider the risk of the portfolio of the investor. Namely, this second group of metrics considers the daily variation the investor's capital, depending on if a long/short position is active. If no positions are active at a certain time, then the daily variation on those periods is zero (theoretically, there is a discount rate that decreases the "value of money" over time, that we will ignore for simplicity purposes).

The following list corresponds to a small set of the existing standard metrics within the first group (i.e. the return of each trade). We consider the return as the profit percentage of a position calculated using the opening and closing prices, after discounting the transaction costs. For simplicity purposes, we assume that the amount of assets traded per position is always equal.

- **Profit Factor:** It is defined as the ratio of the sum of the returns of the profitable trades over the sum of the losses. Values over one are desirable.
- **Payoff Ratio:** The absolute ratio between the average return of a winning trade over the average loss of a losing trade. However this metric has a limitation, since with a very reduced number of winning or losing trades, any of the respective averages may not be significant and lead to misleading results. Suppose a trading system has obtained several successful trades and only three not successful ones (with large losses). The average loss would thus be high leading to a poor payoff ratio, while the trading system was actually producing “good” signals. Nevertheless, this problem is only present for a small number of trades. Scores higher than 1 are also desirable.
- **Average Trade:** The average of the returns obtained when closing a position. Naturally, a value higher than zero is a good indication, though zero would already imply that the system is at least being able to at least cover the transaction costs.
- **Winning Trades:** The percentage of winning trades.
- **Expectancy:** A metric described in [Tharp et al. \(2007\)](#). It tells you how much you should expect to make on average per currency unit at risk. Its formula is given by the average trade return over the absolute average loss of the portfolio. Therefore, a portfolio that achieved a 0.6 expectancy score (which is considered to be a very good value), can be interpreted as “for every unit of currency risked, this system will generate on average 0.6 units of currency”.
- **Total Return:** It is the sum of all the returns (wins and losses included). A total return of 0.05 over a certain period means that the investor would have reached the end of this period with a 5% increase on his capital (assuming the hypothesis that every position had the same number of assets involved)

All these metrics fail to penalise “successful” trades where quite high draw downs during the period of the position were achieved. In order to evaluate the risk of a trading system, one should look to daily (or some other unit of time) variation of the investor’s capital during the given period. In this sense, a return is now the variation of the capital from one unit of time to the next one. If no positions are active at a certain time, the daily variation will be zero. If a long position is active, then a positive return will occur when the asset’s price increases while if a short position is active, a positive return will be considered when that price decreases.

We list some metrics that evaluate the risk of a portfolio obtained using a certain trading system. Once again, we consider that the amount of assets bought and sold per position is always the same. For a more detailed description on portfolio evaluation metrics, we suggest the reading of the survey of [Le Sourd \(2007\)](#).

- **Average Variation:** It is the average of the daily variation of the investor's capital (returns);
- **Value at Risk:** It measures the potential loss in value of a risky portfolio over the given period for a given confidence interval. We consider a 95% confidence level, meaning that this value will be calculated as the 0.05 percentile of the returns of this portfolio. Thus, if the VaR is, for instance, equal to -3% , then with 95% confidence level, this portfolio will not incur in a loss over 3%. It can also be interpreted in the opposite way, as a 5% probability of having a loss over 3%. Having a Value at Risk close to zero is a very good indication regarding the risk of the portfolio. For a very formal, precise and theoretical reading on this metric, we advise the reading of [Peng \(2009\)](#)
- **Tail Value at Risk:** Also described in a very formal way by [Peng \(2009\)](#), this metric measures the expected loss knowing that a loss larger than an amount X has occurred. According to the author, this metric is a more robust and meaningful metric than the Value at Risk. We will choose the Value at Risk as X , meaning that this metric will be the same as the Expected Shortfall. A TVaR of -10% , states that, knowing an extreme loss will occur¹, then in average that loss will be equal to 10%.
- **Sharpe Ratio:** The most generic definition is given by $E[R - R_b]/V[R - R_b]$, where R is the vector of the returns of the portfolio under evaluation while R_b is the respective vector for a benchmark portfolio. Usually, the latter is considered to be a constant risk-free return, thus making $V[R - R_b] = V[R]$. This metric measures the excess return (over that constant benchmark portfolio) per unit of deviation in an investment asset or a trading strategy. For more information regarding the Sharpe Ratio, we recommend the reading of [Lo \(2002\)](#). In our application, we will use a zero constant risk-free return portfolio.

In our application of our proposals to evaluate real trading signals, we will calculate ratios of all these metrics over the respective scores achieved by the optimal signals for several trading policies (note that there are some metrics, mainly regarding the returns of the trades, in which we can not calculate the score for the optimal signals, since they need to account for unsuccessful long/short positions, which will not exist in the optimal benchmark).

¹considering an extreme loss as belonging to the 5% largest losses of the portfolio returns distribution

4.5.3 An Application to Evaluate Real Trading Signals

A set of real trading signals was obtained from an experienced trader and we shall use that opportunity to test the proposed benchmark. By using the benchmark the investor may adapt most evaluation tools to his own trading policy, thus allowing a better analysis of any trading system.

Figure 4.5 shows the trading decisions of a specific trader for the S&P500 index, from the beginning of the year 1990 till 1991, where each decision (an arrow in the graph) corresponds to either selling or buying S&P500 securities.



Figure 4.5: Real Trading Signals for one year period on the S&P500 index. Red arrows correspond to the timings in which assets were sold, while the green ones correspond to the timings in which assets were bought.

In order to know the returns associated to his trade, we need to know the exact timings were every short or long position was opened and closed. Unfortunately, one can not state that every pair of consecutive signals in this figure forms either a long or short position. The reason is the fact that this trader may decide to carry out several successive orders of the same type (e.g. issue a Buy order at time t and issue another Buy at time $t + x$ before the position opened by the first order is closed). Moreover, the investor has traded varying amounts of assets on each order, meaning that the total assets obtained by opening a long position at a certain time t , could be sold separately at different times $t + h_1$, $t + h_2$, etc. This means we had to do some pre-processing of the original data we were given by the trader, by considering at every point in time, a queue of assets that were owned by the investor but not traded yet. The need for this pre-processing results from the fact that our proposed method consists on comparing the performance of a set of positions (created by open and close orders), against the set of ideal positions (defined by the ideal signals).

As an illustrative example, suppose that a short position was opened in the first red arrow of Figure 4.5 issuing a Sell order of 100 assets. This amount of assets needs to be bought at sometime in the future in order to close the short position. Suppose now that in the upcoming green arrow, that correspond to a Buy order, the investor only buys 80 assets. Then, we consider that a short position was opened at the red arrow and closed on the green arrow with an amount of 80 assets. However, there are still 20 assets that need to be bought, meaning that we still need the next buy signal for those 20 assets and close a second short position. This means that in effect we have more than one short position opened at the first red arrow (two if the trader buys the remaining 20 assets all at the same time). We keep moving in our time line, but another red line appears (Sell signal). There are 20 assets that still need to be bought yet the investor wants to reinforce his position by opening another short position with, let's say, 30 assets. After this moment, the investor has still 20 assets from the first red line to buy and now he has added more 30 to that amount. At last, suppose in the upcoming green arrow, the investor buys 90 assets. This amount is enough to fully close the short position opened at the first red arrow and also to close the short position opened at the second red arrow. This means that, after this trade, we consider two more short positions closed, leading to a total of 3 short positions executed during this period. Note that after closing these positions, the trader has now $90 - 50 = 40$ assets in his possession that need to be sold at some point in the future. That means the next red arrows will be used to close long positions opened at the current time until all the 40 assets are sold.

Summarizing, the positions of this illustrative example are:

- Short position opened at the first red arrow and closed at the first green arrow.
Amount of 80 assets;
- Short position opened at the first red arrow and closed at the second green arrow.
Amount of 20 assets;
- Short position opened at the second red arrow and closed at the second green arrow.
Amount of 30 assets

Note that after closing the above positions the trader holds 40 assets, so one or more long positions will still appear in his trading record.

Once we have pre-processed the signals given by the trader, we can check what would be the optimal signals for this interval of time. In order to do that, we have to specify some trading policies. In all the cases, we will consider a transaction cost of 0.5%.

We present four possible trading policies in Table 4.1 and the respective optimal trading signals for each one of them in Figure 4.6.

In the trading policy A, the number of positions is the highest in contrast to their duration which are the shortest of all the four trading policies. The first is explained by the usage of a lower minimum wanted return, that allows more opportunities to successful

	Minimum Wanted Return	Maximum Draw Down Allowed
A	1%	0.5%
B	2%	1%
C	4%	2%
D	4%	4%

Table 4.1: List of trading policies.

trades to occur. However, the investor will not be able to keep a position for a very long time, since the maximum draw down allowed is quite low. This trading policy tends to favour traders who search for a high frequency trade.

The trading policy B seeks for higher returns but also accepts more risk. This will lead to less trades, but longer and more profitable ones. Comparing to the previous one, we can see consecutive short (or long) positions fused, such as the first two short positions in the trading policy A that now form a single one in the new policy. We can also see periods of time now absent of any trading contrarily to the former policy. We are stressing this out, to reinforce the importance of the trading policy of an investor. It certainly leads to different records of optimal trades.

In the last two trading policies, we have increased the minimum wanted return to 4% in both cases, but with two different levels of risk associated. While the first feels more natural, since the investor is looking for high returns assuming a certain and somewhat safe risk, the latter allows a maximum draw down equal to the minimum wanted return. The point here is to stress out the influence of the maximum draw down. Assuming a higher level of risk (i.e. higher maximum draw down allowed), and for the same target return, the investor will not mind waiting a considerable duration of time until he can achieve his target return. Therefore, we see longer positions, including some fusion of consecutive positions.

With these four trading policies we think we cover a considerable set of real world trading scenarios used by traders and investors. Having the given signals (provided by a real trader) and the optimal signals created with different policies, we can now advance to the next stage. At first, we will analyse the given signals using the standard trading evaluation metrics. In a second step, we will analyse the ratios of the scores obtained by the original signals over the ones obtained by the optimal signals.

Table 4.2 shows the scores of all the metrics described before, for the provided signals as well as for the optimal signals obtained for different trading policies, over one year period. The last four pairs of columns pay respect to the scores of the optimal signals for each trading policy as well as the ratio between the score of the trading system being tested over the optimal one.

Let us first analyse the real trading signals (column “Signals”). Overall, it seems to be a decent trading system. The profit factor score indicates the returns of the trades

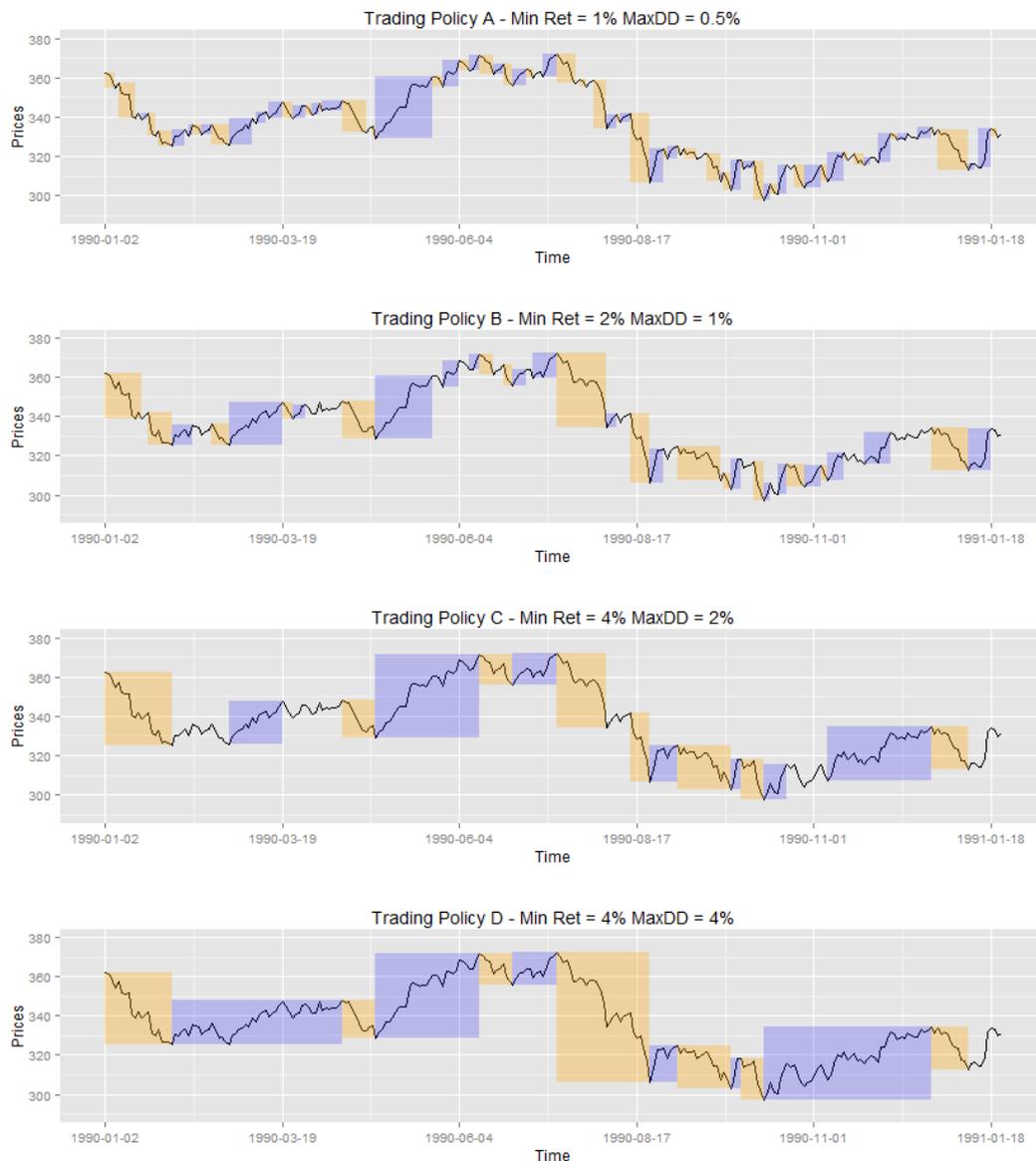


Figure 4.6: Optimal Trading Signals for one year period on the SP500 index.

were 30% larger than the losses. However, the pay-off ratio of this trading system says that a winning trade of this trading system is on average equal to 90% of an absolute loss. It may seem that there is some discrepancy with the previous metric, but if there are more winning trades than losing ones, then both scores are compatible. The average trade return as well as the percentage of winning trades do not have impressive values, though they surpass the minimum values desired (0 and 0.5 respectively). Note that these values include the transaction costs. The result obtained in the Expectancy metric, suggests that for every unit of currency risked, it is expected that the system will generate 0.13 units of currency. In terms of the Total Return, assuming the investor would trade the same number of assets on very position, at the end of the testing period, the investor would have

his capital value increased by 7%. This is a very considerable amount. Typically, there are no bank products that can offer such rates over a one year period, and if there are, they involve large amounts of risk.

		Signals	Opt A	Ratio A	Opt B	Ratio B	Opt C	Ratio C	Opt D	Ratio D
Return	Profit Factor	1.303								
	Payoff Ratio	0.955								
	Average Trade	0.003	0.025	0.106	0.044	0.061	0.073	0.037	0.081	0.033
	Win Trades	0.577	1.000	0.577	1.000	0.577	1.000	0.577	1.000	0.577
	Expectancy	0.128								
	Total Ret	0.070	1.950	0.036	1.533	0.045	1.098	0.063	1.049	0.066
Risk	Average Port	0.001	0.007	0.086	0.005	0.114	0.003	0.168	0.003	0.188
	VaR	-0.017	-0.002	7.350	-0.006	2.656	-0.009	1.967	-0.011	1.505
	TVaR	-0.024	-0.004	6.053	-0.009	2.739	-0.012	1.985	-0.014	1.664
	Sharpe Ratio	0.054	0.839	0.065	0.573	0.095	0.377	0.143	0.306	0.177

Table 4.2: Scores obtained by the trading system and by the optimal benchmark during the year of 1990 of the SP500 index. The Signals column corresponds to the scores of the trading system, while each of the remaining pair of columns are the scores of the optimal benchmark and the respective ratio with the trading system's score.

Evaluating the risk of the orders generated by this trading system, the following can be stated. On average the investor's capital increase 0.1% per day. The values of VaR and TVaR are not so easy to judge. The first one suggests that in 5% of the days, this portfolio value will decrease almost 2%, while the latter says then the average loss of the 5% worst days will be 2.4%. A decrease of this nature during a single day is something serious. Finally, the Sharpe Ratio value is rather modest, suggesting a quite high standard deviation of the returns and therefore some considerable risk.

All this thorough analysis of the given signals has a point. At this moment, we may have gathered some solid information about the performance of this trading system. However, even when using these 10 metrics, it is not easy to argue that this trading system would fit a certain investor profile. Would an investor who prefers to keep positions active for lesser time be interested in this trading system? Or, on the other hand, would an investor who does not mind waiting a long time but expects larger returns be more keen on this system? Moreover, suppose there is a second trading system, with better and worse values across these 10 metrics. How can we choose the best one depending on the trading preferences of an investor?

In order to answer the first two questions, the investor may look at the ratios on Table 4.2 and check whether those values are good enough. For instance, suppose we have an investor who has a a profile that is similar to the trading policy A described on Table 4.1. This means that, this investor is not willing to take barely any risk, but will also be satisfied as soon as return of 1% is achieved. If the investor wants to test the trading system evaluated on this study, then he should look at the Ratio A column and reflect over those values. The average trade would be 10.6% of what would be ideal for this investor as well as the total return that would be 3.4% of the "ideal" value. The average variation in

terms of the portfolio created using this trading system would be 8% of the ideal setting, while the VaR and TVaR would be 7 and 6 times higher than what would be possible, suggesting too much risk for this investor. Finally, the Sharpe Ratio would achieve 6.5% in the same ratio. We claim that with this extra information, the investor may more easily decide if this system is good enough for him.

Note that these values may seem rather low, but achieving a 100% score in any of these ratios is almost impossible for obvious reasons. Achieving, for instance, 20% of what would be the optimal scores is already something to outstanding.

Regarding the third question described before, if the investor has to choose between two trading systems, the ratios we have been using may also be used. In order to illustrate this we generate a second trading system (constructed by artificially applying some perturbations to the original one), where the respective scores are given in Table 4.3. Suppose our investor is now more interested in long-duration positions, searching for very high returns per trade and willing to take a lot of risk in order to get these returns. This means the investor has a trading preference that is similar to the D policy described on Table 4.1.

		Signals	Opt A	Ratio A	Opt B	Ratio B	Opt C	Ratio C	Opt D	Ratio D
Return	Profit Factor	1.215								
	Payoff Ratio	1.051								
	Average Trade	0.010	0.025	0.400	0.044	0.227	0.073	0.137	0.081	0.123
	Win Trades	0.611	1.000	0.611	1.000	0.611	1.000	0.611	1.000	0.611
	Expectancy	0.101								
	Total Ret	0.110	1.950	0.056	1.533	0.072	1.098	0.100	1.049	0.105
Risk	Average Port	0.002	0.007	0.214	0.005	0.301	0.003	0.429	0.003	0.501
	VaR	-0.022	-0.002	11.101	-0.006	3.667	-0.009	2.475	-0.011	2.010
	TVaR	-0.031	-0.004	7.751	-0.009	3.445	-0.012	2.583	-0.014	2.214
	Sharpe Ratio	0.045	0.839	0.054	0.573	0.079	0.377	0.119	0.306	0.147

Table 4.3: Scores obtained by a second trading system and by the optimal benchmark during the year of 1990 of the SP500 index. The Signals column corresponds to the scores of the trading system, while each of the remaining pair of columns are the scores of the optimal benchmark and the respective ratio with the trading system's score.

If we want to choose the best trading system for this investor, we need compare the ratios obtained by each trading system on the D columns (Tables 4.2 and 4.3). We can see that the new trading system has overall considerably higher ratios regarding the Average Trade (3% to 12%), percentage of Winning Trades (58% to 61%), Total Return (7% to 11%) and the Average variation of the investor's capital (19% to 50%). However, slightly worse ratios of VaR (151% to 201%), TVaR (166% to 221%) and Sharpe Ratio (18% to 15%) metrics are observed². Looking at these results, one may infer the following:

- Better ratios in terms of the average return per trade and average variation of the investor's capital, suggest that the new system fits more properly this trader preferences, namely in terms of the target return per trade;

²The closer to 100% the better, since that value implies that the "optimal" value was obtained

- A better ratio in terms of the Total Return also suggests that this system is closer to the optimal return that could be obtained according to this trading policy;
- Worse values in the two metrics that measure the potential loss of the portfolio were observed. However, the usage of these two metrics in this context must be done carefully. The optimal trading system would achieve a VaR of -1.1% . A very safe trading system may achieve this value, suggesting they share the same type of risk, but if that trading system is also producing very small returns, then this system will not be ideal at all for this trading policy. The usage of metrics that pay little concern to the profit must be done with special attention to avoid wrong conclusions;
- In terms of Sharpe Ratio, the second trading system seems to be slightly worse. Since this trading system has achieved a higher average daily variation than the first one, it is only reasonable to assume that the standard deviation of those variations was higher than in the first system. This suggests that the new trading system has incorporated some risk, probably more than what the investor would prefer.

It is up to the investor to look at these ratios and choose the best one. While it is clear that the new trading system produces more desirable returns for someone with this trading policy, it may have incorporated more risk than what is acceptable.

The main advantage of this evaluation approach is that these metrics are already in a “scale” of their trading policy. Not only the scores of two different trading systems may be directly compared, one may also check if the individual scores of each system suggest at all they fit the investor’s policy. For instance, the ratio of the average trade of the original system is really low for this trading policy. This single observation could immediately discard this system for this type of investor that is searching for higher returns.

4.6 Conclusions

In this chapter we have presented a new form of evaluating trading systems. More than looking to the return characteristics of every position generated by a trading system, we have considered a new dimension of information - what would be the ideal decision timings of a perfect trading system given the preferences of a certain investor. This idea aroused from looking at trading as a special form of activity monitoring. Looking at trading within this monitoring framework required several adaptations, the main being: (i) detecting an event as soon as possible may not lead to the best profit; (ii) it is not mandatory to have a continuous duration of time in which starting a long/short position would be profitable. Moreover, when considering the trading problem from this perspective, we have realised that the events we were interested in detecting (long and short positions), should depend on some user-defined preference bias. “What would be a successful long or short position?” This question allowed us to incorporate the notion of a trading policy. Having defined what would be a successful long/short position through the specification of a target return and

a maximum draw down allowed, leads to the definition of which are the events we want to monitor/detect, i.e. what are the ideal moments to open and close the trading positions.

Based on our formalisation of trading as activity monitoring, we have developed two concepts. At first, given a certain trading profile, we have shown how to create an optimal benchmark based on the best timings possible to open every long/short position. Secondly, we have shown how to find periods of time in which opening a position could be successful given a trading policy, i.e. how to find the periods of *positive activity*. The main practical use of latter concept is to evaluate a trading system by its ability to forecast a long/short position in the proper timings. We have listed some potential scoring functions that may be used in this sense, hopefully opening doors and justifying further research. On the other hand, we have tested the first concept (the optimal benchmark) in a real application. This benchmark can be used as an evaluating tool for any investor, allowing him to analyse how close is a certain trading system to what would be optimal for that investor. We have used the ratios between the score of the trading system and the optimal one for a certain investor, where we have shown that is possible and relatively easy to determine if one trading system fits a certain trading policy. We have also seen that these ratios may also be used to compare different alternative trading systems, checking which one fits better some trading policy. We believe that this type of information may be very useful for some investors.

Chapter 5

Conclusions and Future Work

In this chapter we summarise the main conclusions of the work carried out in this thesis, as well as some possible directions for future work.

5.1 Conclusions

We have presented and defined the concept of *Actionable Forecasting*, a task where decisions need to be made based on some forecast of a numeric quantity. We have carried out an extensive analysis between two approaches that deal with actionable forecasting problems: (i) the first, and more standard approach, uses a standard regression tool to obtain the numeric forecasts and then in a second step makes the decisions based on these predictions; (ii) the second uses the deterministic nature of the decision processes we are handling to directly forecast the "correct" decision using classification models. To the best of our knowledge, this analysis is novel, making this a contribution to the scientific community.

Our first experimental comparisons addressed generic tasks. Our analysis involved comparing the alternatives on tasks with different the number of classes/decisions in order to capture any eventual trend that could favour one modelling approach for a higher number (or lower) of classes. Our experiments also involved a quite diverse and extensive set of modelling tools to exclude any eventual dependency of our conclusions on some particularities of the tools.

Before directly comparing both modelling approaches, we have considered some potential theoretical problems of the approach based on classification models, where we have tested the usage of cost-benefits matrices in order to overcome them. Namely, we have considered that by ignoring the intermediate numeric variable, the classification models would be losing the ordering information of the classes, potentially increasing the severity of their mistakes. We have tested and observed that by using cost-benefit matrices, the performance of the several classification models has increased. We also observed that this increase of performance would be stronger the higher the number of classes/decisions, which is consistent with our initial hypothesis.

Regarding the direct comparison of the two alternative approaches to actionable forecasting we have carried out different types of studies in order to enhance the robustness of our conclusions. We believe we have gathered enough evidence to state that overall the classification modelling approach can outperform the more frequently used 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.

Having defined the general concept of actionable forecasting we have then moved into the analysis of a specific instance of these problems: financial trading decisions. More specifically, we have studied applications where a forecasting model tries to anticipate the future evolution of the prices of some financial assets and then a trading decision needs to be made, based on these predictions. This instance of actionable forecasting has some characteristics that make it quite different from the generic tasks studied before. Namely, we now are addressing a task based on data that is ordered by time (time series data), and our decisions are rather unbalanced, with the more important decisions being rare. Given these differences, we could not assume that the conclusions of the generic study would hold on the trading problem.

Similarly to the generic problem, we have analysed the possible limitations of each modelling approach, but now for trading tasks. It is well known that most the modelling tools struggle to model unbalanced data sets. In order to deal with this problem, we have considered re-sampling our data sets using the Smote algorithm. We have tested this feature and observed that, overall, even though the models (classification and regression) could properly detect more often the important trading actions, they have also incorporated too much risk leading to serious economic losses. A few percentage of all the models (classification and regression) were able to improve their results by using this re-sampling strategy. We have also tested the usage of cost-benefit matrices to teach the classification models the danger of confusing a selling order with a buying one. We have observed once again, that using this feature improves the performance of several classification models, though in a less evident way than in the generic tasks.

After the analysis of these additional methods that could improve the performance of the models we have finally compared the two approaches (classification and regression) on this particular instance of actionable forecasting. We have used 12 different companies to increase the robustness of our comparisons, and have again considered a large set of modelling tools and parameter variants. We have gathered enough information to conclude that there is no statistically significant difference between both approaches in the context of these financial trading decision problems.

We have presented a third contribution on the thesis. Again in the context of financial

trading, we have proposed a new perspective for evaluating trading systems. We have proposed a formalisation of the financial trading based on the existing framework of activity monitoring. Using this formalisation we have shown how to find the optimal signals with respect to a user-defined trading policy. These ideal signals allowed us to propose an optimal benchmark that is adapted to the user preferences (in terms of target return per trade and the maximum draw down allowed). We claim that this benchmark can be very useful to compare any trading system against it. We have described how to do it and presented a real application (with real data and real trading signals). We claim that this form of evaluating and comparing the performance of some set of trading systems is more informative to potential investors as it allows them to match these systems against the ideal performance according to their own trading preferences. To the best of our knowledge this is a novel way of evaluating trading systems.

Finally, based on the proposed formalisation of financial trading, we have defined the concept of positive activity within these tasks. These are periods of time where issuing a signal is rewarding according to the trader's preferences. This concept provides the bases for defining scoring functions that can be used to characterise/evaluate the signals of any trading system in terms of how far they are from the ideal timings. Moreover, these scoring functions also have potential to be used in other concepts like for instance in terms of trying to forecast their future value and use these predictions for trading.

5.2 Future Work

Regarding the comparison of both the classification and regression modelling approaches to Actionable Forecasting tasks, we claim we have covered most of the key parts: (i) we have considered the limitations of each approach, by considering re-sampling and cost-benefit matrices; (ii) we have considered a set of metrics that cover any potential interest of the user (Accuracy, Utility Score, Recall, Precision); (iii) we have conducted distinct types of analysis that altogether lead to extensive and robust conclusions; (iv) we have considered generic non-temporal tasks and temporal tasks (trading only). This last point is, perhaps, the only one that can be strengthened. As some future work, we could open our analysis for generic temporal tasks or generic non-temporal but heavily unbalanced for instance. This would cover almost any actionable forecasting task setup. Still, an exception that was not covered in this thesis are problems where the decision, given a numeric forecast, is non-deterministic. Whether a similar study could be carried out for this other type of decision problems is another topic for future research.

Regarding our proposal for an optimal benchmark and positive activity for evaluating trading systems, there is some future work to be considered. Concerning the optimal benchmark, the main path to follow from this moment is to focus on the creation of new metrics. The same way that metrics such as the Sharpe Ratio analyse the excess return over a null benchmark at the cost of an increase in terms of risk, one can also think of

metrics that analyse how close is the return to the optimal one and at which costs in terms of excess risk over the optimal one. Moreover, these decisions would depend on the trading policy of the investor, making these new metrics potentially more interesting to investors. In terms of the positive activity periods, we believe that some research should be carried out in terms of finding the most informative scoring functions and the characteristics of the distribution of the respective scores. The possibility of using these scoring functions in the context of prediction models is also a potentially interesting avenue for future research.

Appendices

Appendix A

Actionable Forecasting - Generic Tasks

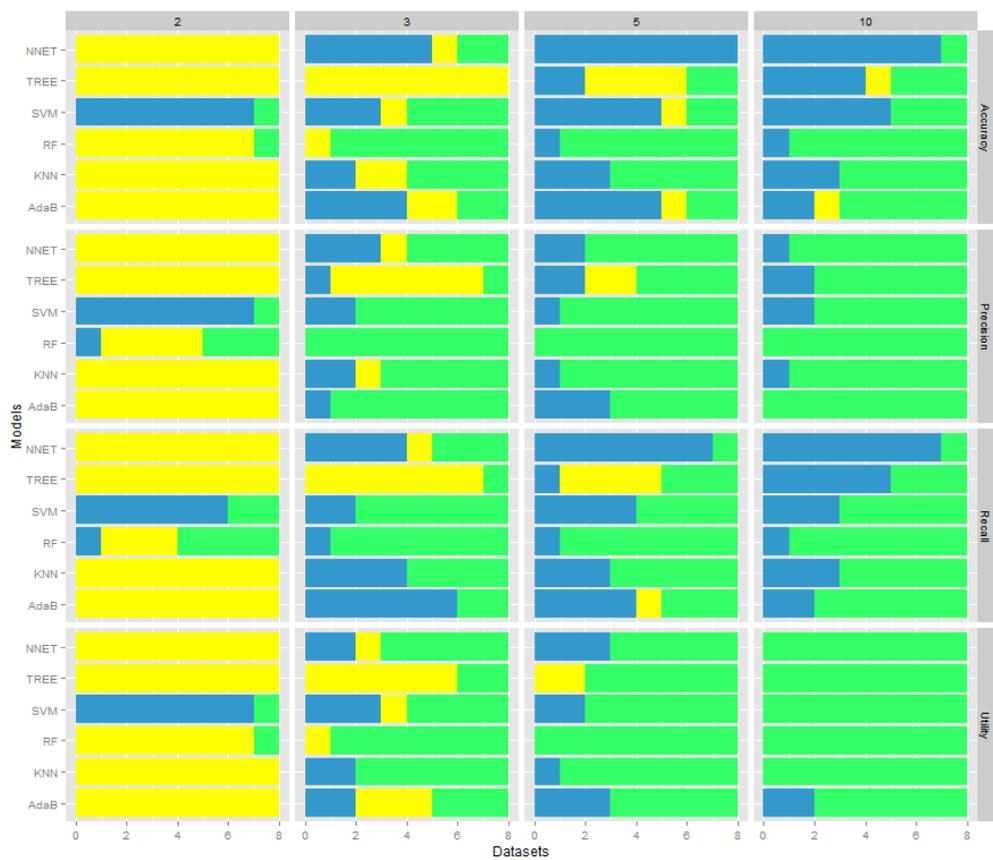


Figure A.1: Segmented by type of model and by metric, the median score of each group is directly compared (Classification without costs vs Classification with costs). Since there are 8 data sets in each segment, then there are 8 results per segment. Each bar shows the results for each segment, where each one of the three colours is associated to a type of win (without costs - light blue, draw - yellow, with costs - light green) The length of each colour describes the number of times that type of win occurred.

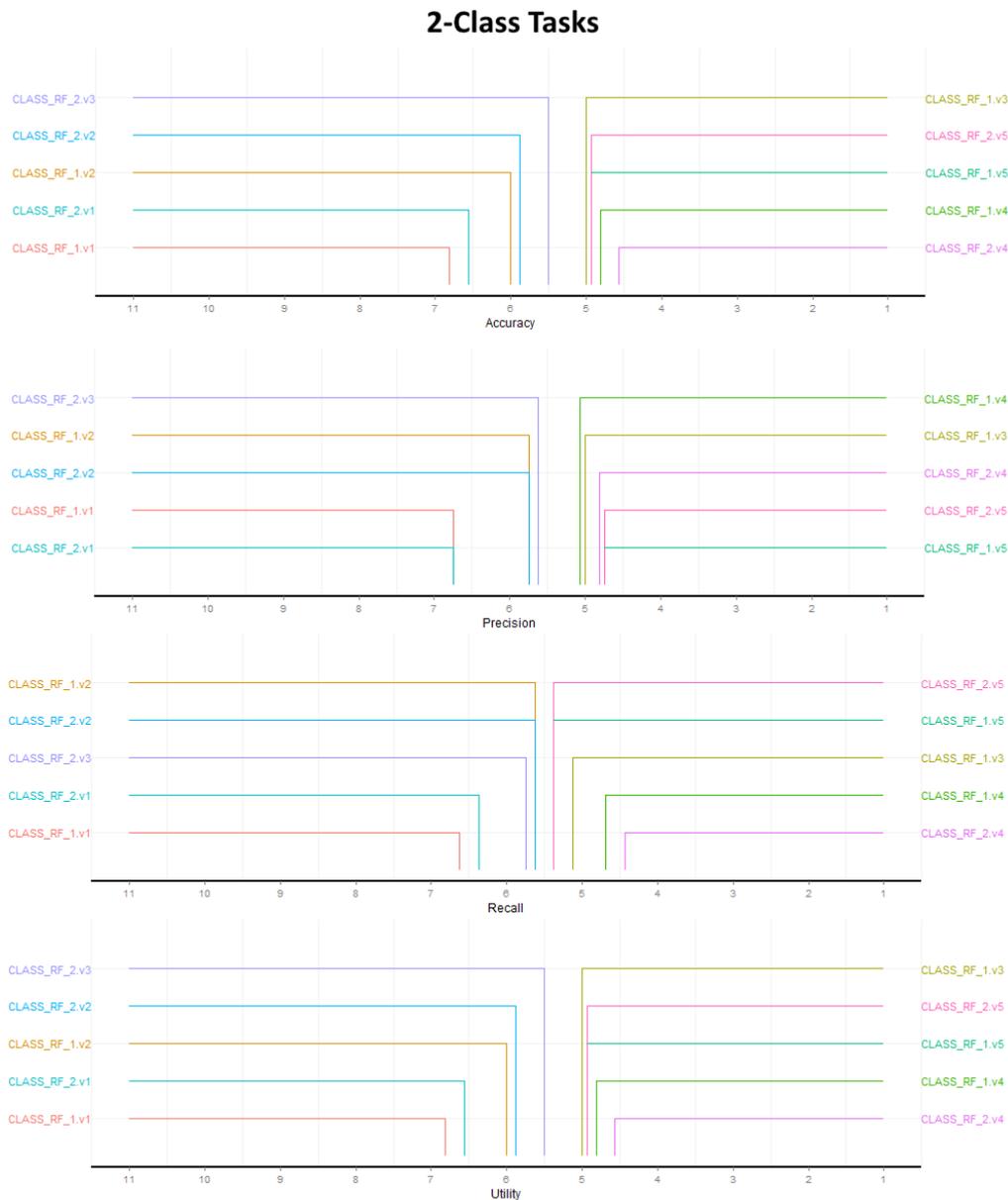


Figure A.2: The top five average ranking model variants of both modelling approaches (Classification without costs and Classification with costs) are forming a new set of variants, and their average rankings are re-calculated within this new set of models for tasks with 2 classes only. Each model is thus given an average ranking (x axis). The presence of at least one black line implies that the Friedman's 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's statistical test.

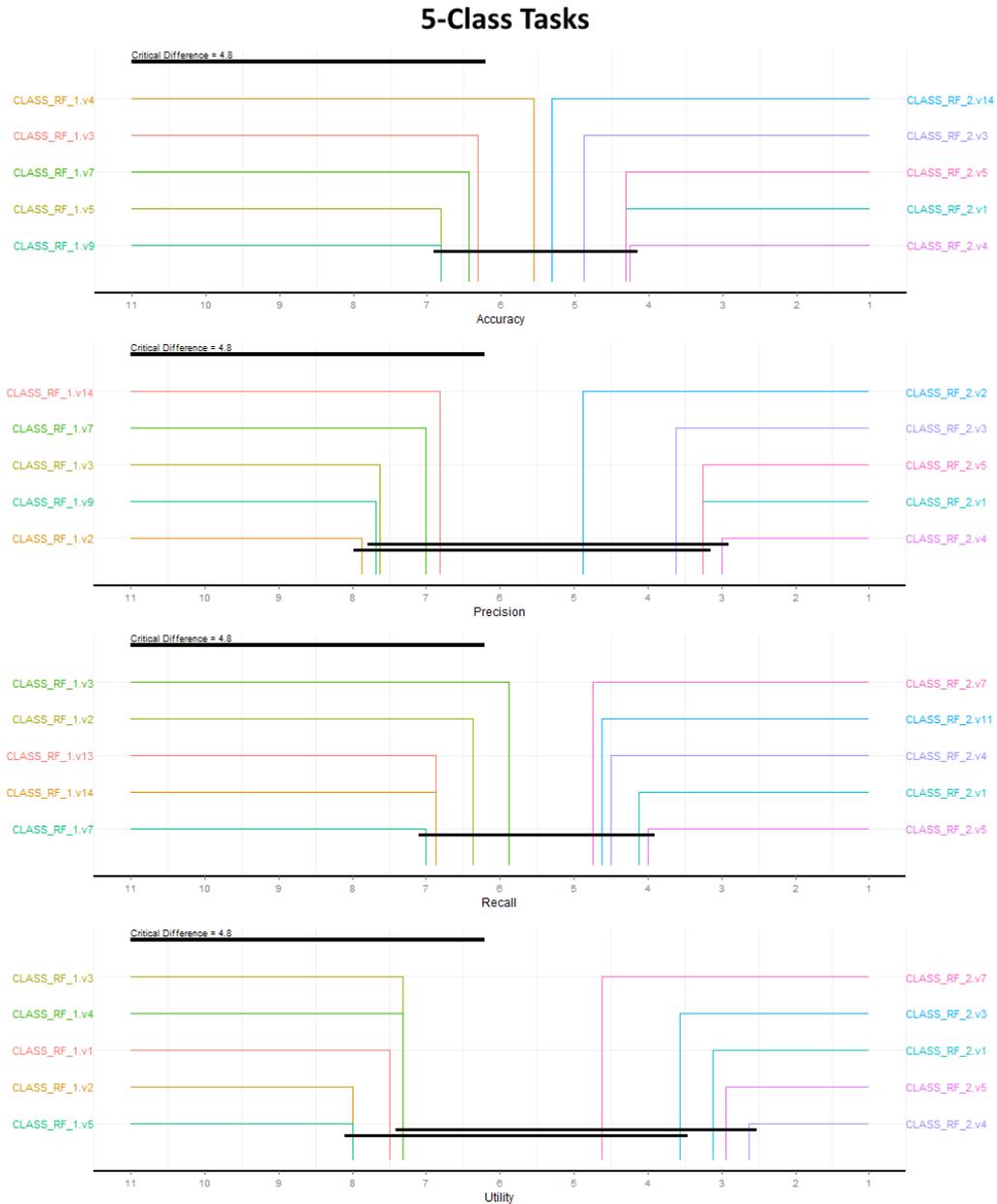


Figure A.3: The top five average ranking model variants of both modelling approaches (Classification without costs and Classification with costs) are forming a new set of variants, and their average rankings are re-calculated within this new set of models for tasks with 5 classes only. Each model is thus given an average ranking (x axis). The presence of at least one black line implies that the Friedman’s 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’s statistical test.

Appendix B

Actionable Forecasting - Trading Tasks

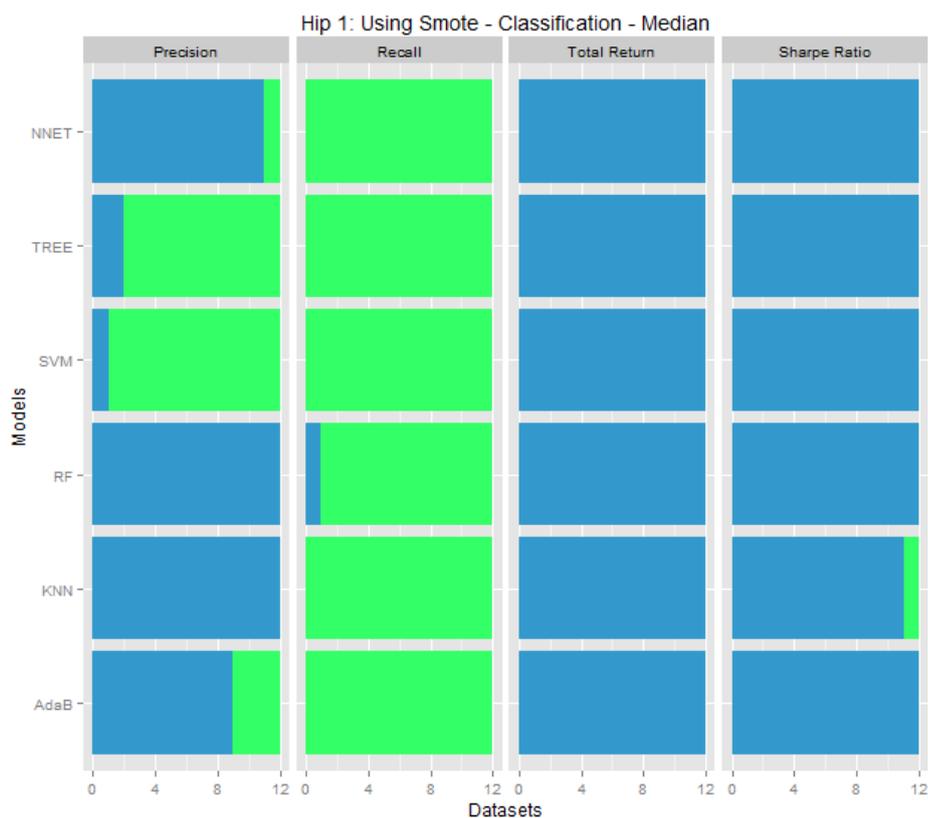


Figure B.1: Segmented by type of model and by metric, the median score of each group is directly compared (Classification without SMOTE vs Classification with SMOTE). Since there are 12 data sets in each segment, then there are 12 results per segment. Each bar shows the results for each segment, where each one of the three colours is associated to a type of win (without SMOTE - light blue, draw - yellow, with SMOTE - light green) The length of each colour describes the number of times that type of win occurred.

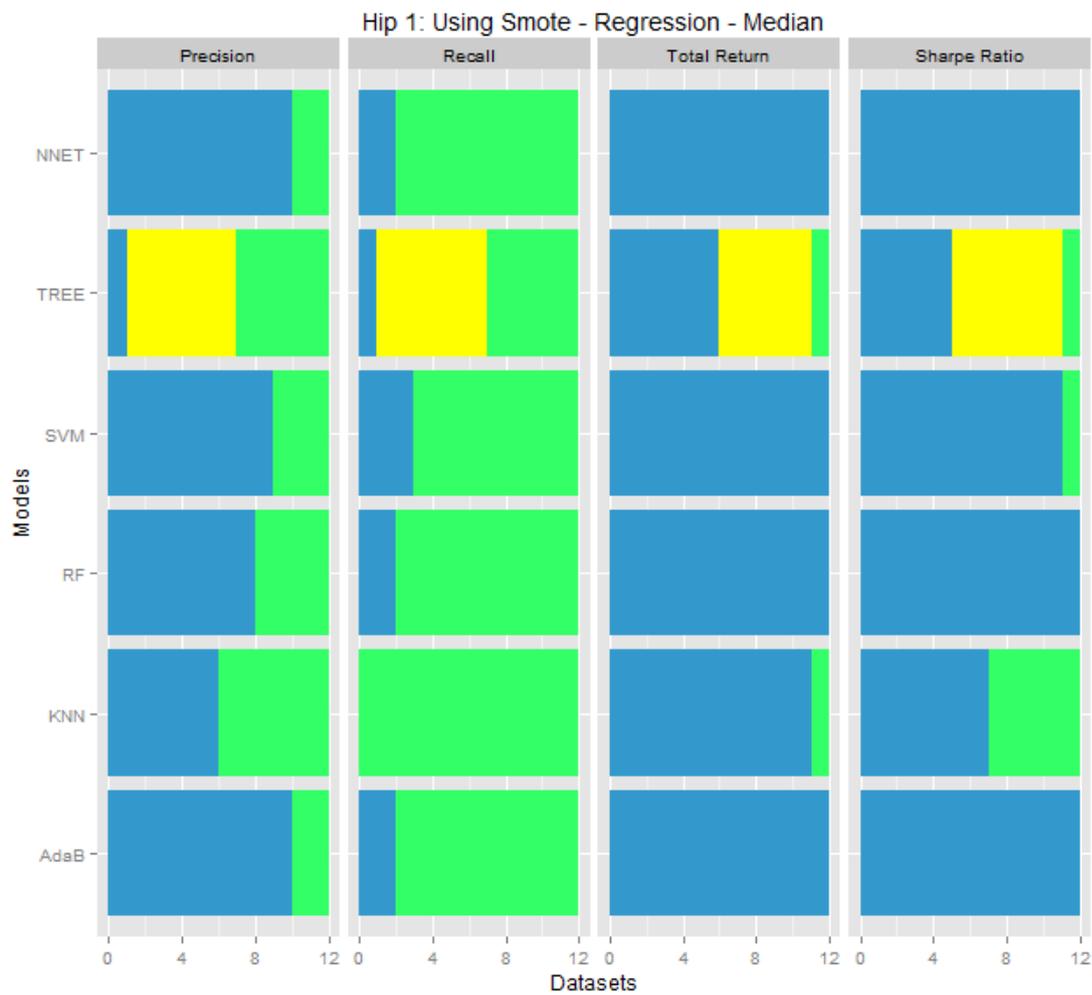


Figure B.2: Segmented by type of model and by metric, the median score of each are directly compared (Regression without SMOTE vs Regression with SMOTE). Since there are 12 data sets in each segment, then there are 12 results per segment. Each bar shows the results for each segment, where each one of the three colours is associated to a type of win (without SMOTE - light blue, draw - yellow, with SMOTE - light green) The length of each colour describes the number of times that type of win occurred.

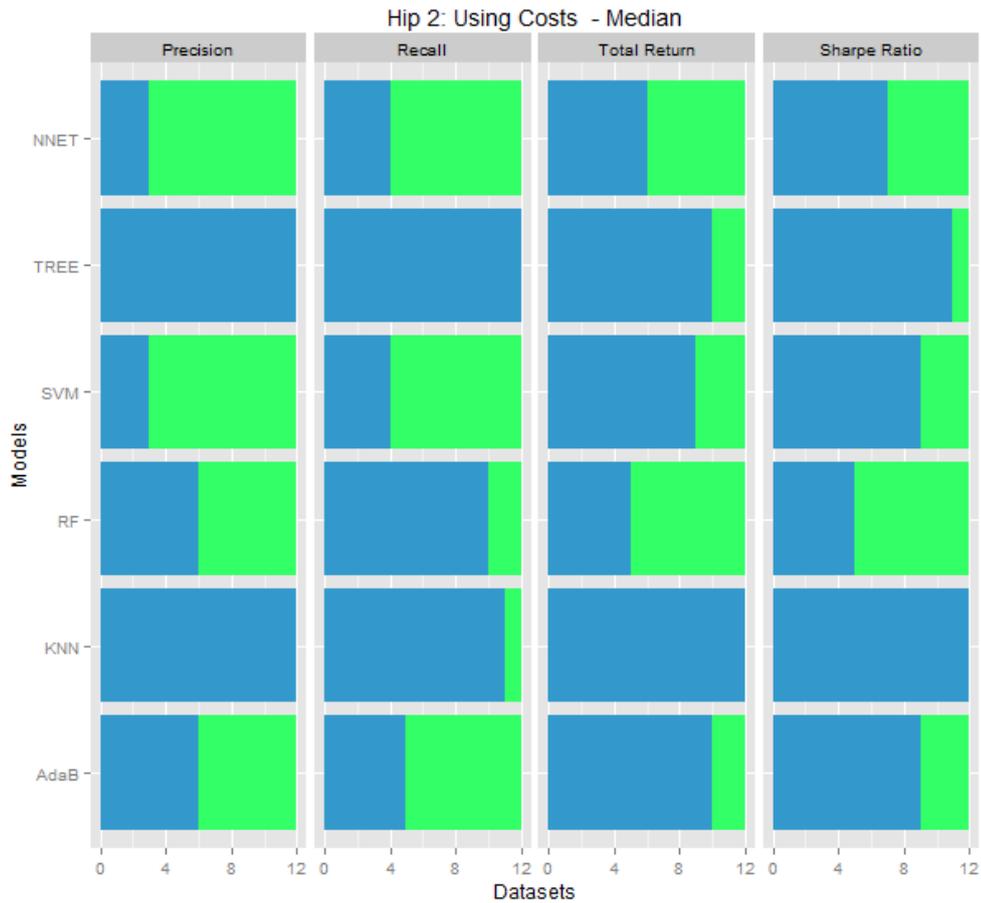


Figure B.3: Segmented by type of model and by metric, the median score of each group is directly compared (Classification without costs vs Classification with costs). Since there are 12 data sets in each segment, then there are 12 results per segment. Each bar shows the results for each segment, where each one of the three colours is associated to a type of win (without costs - light blue, draw - yellow, with costs - light green) The length of each colour describes the number of times that type of win occurred.

Bibliography

- Ahmed, N. K., Atiya, A. F., Gayar, N. E., and El-Shishiny, H. (2010). An empirical comparison of machine learning models for time series forecasting. *Econometric Reviews*, 29(5-6):594–621.
- Arlot, S., Celisse, A., et al. (2010). A survey of cross-validation procedures for model selection. *Statistics surveys*, 4:40–79.
- Atsalakis, G. S. and Valavanis, K. P. (2009). Surveying stock market forecasting techniques – part ii: Soft computing methods. *Expert Systems with Applications*, 36(3, Part 2):5932 – 5941.
- Bailey, D. H. and Lopez de Prado, M. (2012). The sharpe ratio efficient frontier. *Journal of Risk*, 15(2):13.
- Branco, P., Torgo, L., and Ribeiro, R. (2015). A Survey of Predictive Modelling under Imbalanced Distributions. *ArXiv e-prints*.
- Chang, P.-C., Fan, C.-Y., and 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., and 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.
- Desai, A. and M Jadav, P. (2012). An empirical evaluation of ad boost extensions for cost-sensitive classification. *International Journal of Computer Applications*, 44(13):34–41.
- Fawcett, T. and Provost, F. (1999). Activity monitoring: Noticing interesting changes in behavior. In Chaudhuri and Madigan, editors, *Proceedings on the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 53–62, San Diego, CA.

- Fernandes, M. (2002). Tecnicas de ia aplicadas a previsão de series temporais financeiras.
- Ferruz, L., Pedersen, C., and Sarto, J. L. (2006). Performance metrics for spanish investment funds. *Derivatives Use, Trading & Regulation*, 12(3):219 – 227.
- Folger, J. (2012). How to leverage a performance report. *Futures: News, Analysis & Strategies for Futures, Options & Derivatives Traders*, 41(5):34 – 36.
- Gençay, 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., and 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.
- Kao, L.-J., Chiu, C.-C., Lu, C.-J., and 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.
- Le Sourd, V. (2007). Performance measurement for traditional investment. *Financial Analysts Journal*, 58(4):36–52.
- Lee, Y.-H., Wei, C.-P., Cheng, T.-H., and Yang, C.-T. (2012). Nearest-neighbor-based approach to time-series classification. *Decision Support Systems*, 53(1):207 – 217.
- Liaw, A. and Wiener, M. (2002). *Classification and Regression by randomForest*.
- Lo, A. W. (2002). The statistics of sharpe ratios. *Financial Analysts Journal*, 58(4):36–52.
- Lu, C.-J., Lee, T.-S., and 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. and 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., and Huang, D.-S. (2012). Multiple costs based decision making with back-propagation neural networks. *Decision Support Systems*, 52(3):657 – 663.
- McCrum-Gardner, E. (2008). Which is the correct statistical test to use? *British Journal of Oral and Maxillofacial Surgery*, 46(1):38–41.
- Meyer, D., Dimitriadou, E., Hornik, K., Weingessel, A., and 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.
- Pardo, R. (2011). *The Evaluation and Optimization of Trading Strategies*. Wiley Trading, Wiley.
- Pav, S. E. (2014). Notes on the sharpe ratio.
- Peng, J. (2009). Value at risk and tail value at risk in uncertain environment. In *Proceedings of the 8th International Conference on Information and Management Sciences*, pages 787–793.
- R Core Team (2014). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.
- Ridgeway, G. (2013). *gbm: Generalized Boosted Regression Models*. R package version 2.1.
- Ridgeway, G., Southworth, M. H., and RUnit, S. (2013). *Package ‘gbm’*.
- Sermpinis, G., Dunis, C., Laws, J., and 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. and Ghosh, J. (1995). Ridge polynomial networks. *IEEE Transactions on Neural Networks*, 6(3):610–622. cited By 64.
- Sokolova, M. and Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing Management*, 45(4):427 – 437.
- Teixeira, L. A. and 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.
- Tharp, V. K., Chabot, C., and Tharp, K. (2007). *Trade your way to financial freedom*. McGraw-Hill.
- Torgo, L. (2010). *Data Mining with R, learning with case studies*.
- Torgo, L. (2013). *An Infra-Structure for Performance Estimation and Experimental Comparison of Predictive Models*.
- Torgo, L. and Dhar, V. (2004). Trading as activity monitoring. Technical report.
- Torgo, L. and Gama, J. (1997). Regression using classification algorithms. *Intelligent Data Analysis*, 1(1–4):275 – 292.
- Torgo, L., Ribeiro, R. P., Pfahringer, B., and Branco, P. (2013). Smote for regression. pages 378–389.

- Venables, W. N. and Ripley, B. D. (2002). *Modern Applied Statistics with S*. New York, fourth edition. ISBN 0-387-95457-0.
- Weiss, G. M. (2004). Mining with rarity: a unifying framework. *ACM SIGKDD Explorations Newsletter*, 6(1):7–19.
- Weiss, G. M. (2005). Mining with rare cases. In *Data Mining and Knowledge Discovery Handbook*, pages 765–776. Springer.
- Wilder, J. (1978). *New Concepts in Technical Trading Systems*. Trend Research.