Forecasting interest rate movements of the 10-Year Bund for investment purposes by the use of classification algorithms

A Master Thesis Written
by

Robert de Veer
(999656)

Supervised by:
Prof. Dr. B.J.M. Werker

Final Version
28 March 2008

"What makes a data set interesting is not only its size but also its complexity,
where complexity can include such considerations as:
high dimensionality, a mixture of data types, non-standard data structure, and,
perhaps most challenging,
nonhomogeneity;
that is, different relationships hold between variables
in different parts of the measurement space"

Breiman, 1984

Acknowledgements

I would like to thank Prof. Dr. Bas Werker as my thesis supervisor for his support and helpful remarks to keep me on track while writing this Master Thesis. In addition I would like to thank Dr. Frank Meijer, Head Quant Desk AEGON Asset Management (AEAM), for giving me the opportunity to write this Master Thesis at AEGON. I'am most thankful for the daily support of Drs. Robert-Jan van der Mark who made my internship at AEAM both great fun and informative. In this perspective I cannot forget to say thank you for being my colleagues to the rest of the Quant Desk: Eric, Marijn, Niek, and of course Yvo. I wish you all good luck in your careers and personal lives.

With the completion of this Master Thesis comes also (almost) the end of my life as student. My parents are the persons to thank for making it possible to develop myself as human being. I'am very grateful to have you as my father and mother and that you created the environment for me to be who I'am now. Thank you! The last person I would like thank is my lovely girlfriend Marjolein. Dear, you make it worth to live our lives to the fullest.

Contents

Chapter 1. Introduction

In asset management the focus of institutional investors is shifting from beta-performance to alpha-performance. The traditional beta-performance is obtained by following benchmarks like the AEX and the 10-Year Bund Index by taking long-positions in these markets. Clients of institutional investors increasingly desire an outperformance on those benchmarks. The outperformance by institutional investors is achieved by investing a percentage of a portfolio in short-positions and using leverage to accelerate gains (but also losses!) on investments bets. These developments are also experienced by AEGON Asset Management B.V., the Dutch investment division of AEGON N.V.

AEGON N.V. is an international insurance company and mainly active on the market of life insurance, pension plans, and individual investment products. The Netherlands, Great-Britain, and the United States are the most important markets for AEGON. Worldwide AEGON N.V. manages over €360 billion, of which €60 billion is managed AEGON Asset Management B.V. The investment style adopted by AEGON Asset Management is called 'enhanced active'. Goal is to beat the benchmark after transaction costs within a strict framework of risk control. In practice 'enhanced active' means that portfolio managers have the freedom to take small restricted positions different from the benchmark. The investment process is designed such that relative allocations at all levels within the portfolios can be taken. Risk control is present by the reduction in size of relative allocations and the incorporation of active investment decisions.

One the most important tools of asset managers to gain alpha-performance are forecast models. Successful forecast models enable them to take profitable positions in all kinds of financial assets. In most forecast models, available in the literature and used in practice, at any observation all (selected) input variables are taken into account. In other words, the influence of all input variables on a forecast is considered. Whereas one might argue that not all variables should be taken into account at all times. This can very well be illustrated from the viewpoint of financial markets. An asset manager probably will not look at the same economic data under all circumstances. In fact, it is more likely that the asset manager makes an ordering of economic data relevance. This ordering will change over time when market conditions change. E.g. in highly volatile markets the asset manager might first look at for example implied volatilities, whereas in relative quiet markets macroeconomic data could be leading variables in investment decisions. In statistics such relations between variables are called non-linear relations. One of the types of models that is designed to reveal non-linear relations in data are called classification trees. As the name suggests data is ordered in a tree sequence. At the end of a path followed through the tree the model gives a classification label as output. In this thesis we will look at the use of classification trees in finance. More specific: "Are classification models useful in forecasting the 10-Year Bund Interest Rate movements for investment purposes?"
To answer our research question we will first look at four classification tree algorithms in Chapter 2. Within this chapter we will discuss binomial trees in which splits in the tree are only binomial, and multiway trees in which each node has as many splits as possible output classes. In Chapter 3 the empirical framework of this thesis will be described. Also a rough selection of suitable

classification models will be made. The next step is to link the use of classification trees to a Bund portfolio performance which is done in Chapter 4. In this fourth chapter sensitivity analysis with respect to portfolio performances is done and the most robust and highest performing classification trees are determined. The consistency of the selected classification model is evaluated in section 2 of Chapter 4. The selected models are evaluated through the performance and robustness over the second half of 2007 and January 2008 (this period was not included in research up to then, therefore we can regard it as an out-of-sample test). Chapter 5 will conclude.

Chapter 2. The theoretical framework

Section 1. Classifying data

§ 2.1.1. A classifier or classification rule

Predicting the most likely condition of an event to occur at a certain time given current information is a problem for which one often wants to find an answer. Given a set of predictors is it possible to say something about a condition today or in the future? Examples are the identification of a disease based on a number of medical symptoms (i.e. a condition today) and forecasting the type of weather the day after tomorrow at a certain place based on different weather conditions today at different places (i.e. a condition in the future). One can refer to the different conditions of an event as classes. The above problem more formally, given a set of variables one wants to have a systematic way of predicting what class a case is in. Such a systematic way is called a *classifier* or a *classification rule*.

Let's describe the above problem more mathematically. Suppose a prediction can be categorized as belonging to a class $j \in J = \{1, 2, \ldots, J\}$ The set of (possible) predictor values can be denoted by X such that every vector $\mathbf{x}$ is a different observation in the set of predictor values. "A systematic way of predicting class membership is a rule that assigns a class J to every measurement vector $\mathbf{x}$ in X. That is, given any $\mathbf{x} \in X$, the rule assigns one of the classes j to $\mathbf{x}$." (Breiman et al., 1984). Define $d(\mathbf{x})$ as a function on X such that to every $\mathbf{x}$ a class j is assigned, thus $d: X \rightarrow J$. Such a function $d(\mathbf{x})$ is called the *classifier* or *classification rule*. It is also possible to define a classifier by defining A_j as the subset of X on which $d(\mathbf{x}) = j$, that is $A_j = \{\mathbf{x} \mid d(\mathbf{x}) = j\}$. Then the classifier is a partition of X into J disjoint subsets $A_1, A_2, \ldots, A_J$, $X = \bigcup_j A_j$ such that for each $\mathbf{x} \in A_j$ the predicted class is j.

§ 2.1.2. The learning sample

Often one wants to construct a classifier based on available observations that are likely to have the same relation between predictor values and a prediction. In the literature such a set of available information is called the *learning sample*. It implies the assumption that another set of predictor values will have the same distribution as the observations in the learning sample. The learning sample contains N observations of each predictor variable together with the actual class of the prediction variable corresponding to the n^{th} observation (vector). A more formal definition is given by Breiman et al.: "A learning sample consists of data $(\mathbf{x}_1, j_1), (\mathbf{x}_2, j_2), \ldots, (\mathbf{x}_N, j_N)$ on N cases where $\mathbf{x}_n \in X$ and $j_n \in J$, $n = 1, 2, \ldots, N$. The learning sample is denoted by L; i.e., $L = \{(\mathbf{x}_1, j_1), (\mathbf{x}_2, j_2), \ldots, (\mathbf{x}_N, j_N)\}$."

Important to note is that there are different types of predictors. Two different types are distinguished in the literature: *ordered variables* and *categorical variables*. Ordered predictors are real numbers and categorical predictors take values in an unordered set.

§ 2.1.3. The purpose of using classification procedures

The purpose of this thesis is to uncover a possible predictive structure for movements of the 10-Year Bund interest rate. Hence we are trying to get an understanding of what variables or interactions of economic data drive these movements. The classification procedures we will consider in this thesis are called *classification tree algorithms*. In section 3 we will get into deeper detail about these algorithms. Besides quantitative criteria regarding the outcomes of the classification trees a more qualitative criterion should be applied to our problem. Namely, "an important criterion for a good classification procedure is that it not only produces accurate classifiers (within the limits of the data) but that it also *provides insight and understanding into the predictive structure of the data*." (Breiman et al., 1984). In this thesis we only consider classifiers based on variables separately and disregard classifiers that permit linear combinations of different variables. The reason is twofold: (i) to limit the possible classifiers; and (ii) there is no reason to assume that linear combinations of financial data make sense in our case; even if there are reasonable linear combinations present these are very likely to already exist as a separate variable (e.g. nominal interest rate – inflation rate ≈ real interest rate). This last argument implies that before we apply an algorithm to find a classifier, we should carefully look at the variables contained in our learning sample, so the real interest rate should be in our learning sample together with the nominal interest rate and the inflation rate.

§ 2.1.4. Misclassification rates

Classifying data comes together with some other questions: 'How precise is a classification rule and how can we estimate its precision?' Therefore we introduce the notation $R^*(d)$ which denotes the *misclassification rate* of a classification procedure. Of course we cannot say what the exact value of $R^*(d)$ is, but we can estimate it. Two estimates of the misclassification rate are often used.

The first one we discuss is based on the intuition that it makes sense to exclude a random selection of the learning sample, determine a classification rule with the remainder of the learning sample, and use the excluded sample to estimate the misclassification rate of the classification rule. Call the remainder of the learning sample L_1 and the out-of-sample set L_2, thus $L = L_1 + L_2$; note that it is assumed that by randomly selecting L_2 from the learning sample, L_2 has the same distribution as L_1. The misclassification rate can be estimated, called *independent test sample estimate*, by looking how often a case in L_2 belonging to class j is not classified as a class j by the classification rule that is based on L_1. Now we can define $R^{its}(d)$ as the independent test sample estimate of the probability that d will misclassify a new sample drawn from L_2, i.e. $R^{its}(d) = [P(d(\mathbf{x}) \neq y \mid L_2)] = E[R^*(d)]$.

A second method for estimating $R^*(d)$ is called *V-fold cross validation*. Divide the learning sample L in V subsets (denoted by $L_1, L_2, \ldots, L_V$) of nearly equal size. For every v, $v = 1, 2, \ldots, V$, apply an algorithm for constructing a classifier with $L - L_v$ as learning sample, and let $d^{(v)}(\mathbf{x})$ be the resulting classifier. With $N_V = N/V$ the number of cases in L_V we can define $R^{ts}(d^{(v)}) = \frac{1}{N_V} \sum_{(\mathbf{x}_n, j_n) \in L_V} 1_n(d^{(v)}(x_n) \neq j_n)$, as a test sample estimate for $R^*(d^{(v)})$. If the number V is taken large then each of the V classifiers is determined using a learning sample of size $N(1 - 1/V)$, which is nearly as large as L. Cross-validation assumes that the classifiers $d^{(v)}$, all constructed with learning samples almost as large as L, have misclassification rates $R^*(d^{(v)})$, which are nearly equal to $R^*(d)$. This is called the 'stable' assumption for cross-validation. Following the above procedure for all $v \in V$ the V-fold cross-validation estimate $R^{cv}(d)$ can be stated as $R^{cv}(d) = \frac{1}{V}\sum_{v=1}^{V} R^{ts}(d^{(v)})$ = E[R*(d)].

The independent test sample estimate is the computational most efficient method to estimate the misclassification rate, because a classifier has to be constructed just once. If a learning sample is large enough this method of estimation is preferred, i.e. if in the learning sample near 1000 observations for each class *j* are observed (according to Breiman et al.). V-fold validation has as advantages that it makes effective use of all observations available and that it gives useful information about the stability of the classifier. A classifier with misclassification rates that do not differ much for each subset V, is much more reliable than one for which they do differ much.

Section 2. General tree building procedures

Our framework so far: We have a set of past experiences, called *learning sample*, and a variable to be forecasted that is a set of classes. Given the learning sample we want to find a systematic way, called *classifier* or *classification rule*, of predicting what class a case is in. The classification procedures we discuss in this thesis are called *classification tree algorithms*. They have in common that they are based on minimizing *misclassification rates*.
In this section the general tree building procedures will be discussed.

§ 2.2.1. Steps in building a classification tree

In the literature two types of classification trees are considered: *binary trees* and *multiway trees*. Binary trees are constructed by repeated splits of subsets of the learning sample X into only two mutually exclusive subsets, one starts with splitting the set X itself. Each split is based on one of the variables in the set X. Mulitway trees are constructed by repeated splits of subsets of X into two or more mutually exclusive subsets. The number of mutually exclusive subsets by which a split is divided, depends on both the number of classes J and the way an algorithm is constructed. Splitting of a subset stops if a *stopping rule* (will be discussed in §2.2.3) is satisfied or in case the subset is *pure*, meaning that all y_i in the subset belong to the same class j. Subsets that have no further splits are called *terminal subsets*; a class is assigned to each terminal subset. The entire construction of a classification tree includes three steps:

1.) The split selection
2.) The decision whether to continue splitting a subset or to stop
3.) The assignment of each terminal node to a class

In theory every multiway tree can be redrawn by a binary tree, since every split in a multiway tree can be replicated by a sequence of binary splits. Nevertheless as we will discuss later on, multiway trees do have some advantages above binary trees. The idea is that depending on the applied splitting method different non-linear relations are revealed. One should keep in mind that selecting a classification tree algorithm is more related to what one wants to forecast and what type of dataset is involved, than selecting just the algorithm that minimizes some misclassification rate.

§ 2.2.2. Split selection variable

The split selection procedure is the part that distinguishes different classification tree algorithms from each other the most. In general split selection involves two steps:

1.) The selection of the split variable
2.) The selection of the split point of a variable

Which of the two steps to take first is dependent on the chosen algorithm. For example, with CART (Breiman et al, 1984) the 'best' split for each variable is selected first, then the variable with the 'best' split is chosen as the to be split variable; whereas with FACT (Loh and Vasichsetakul, 1988) first the 'best' split variable is chosen and then how to split this 'best' variable. The selection how to split a variable is also dependent on the choice whether to build a binary tree or a multiway tree.

Splits can have two forms:

1.) $X \leq c$, with $c \in (x_{min}, x_{max})$; if X is an ordered variable
2.) $X \in A$, with A a non-empty subset of the values taken by X; if X is a categorical variable

Of course, each split selection procedure has its advantages and disadvantages. These and the different split selection procedures will be discussed in the next section.

§ 2.2.3.Stop splitting

As already mentioned the splitting of a subset of X stops if the subset is pure or a stopping rule is satisfied. The reason for stop splitting in case a subset of X is pure is straightforward; a subset is pure when all y_i in the subset belong to the same class j and therefore there is no need for further splits, i.e. there is a 0% misclassification rate.

There are two sorts of stopping rules. Splitting rules of the first sort are very simple and can directly be observed when looking at a subset in a node. Three (primitive) stopping rules of this first type are: (i) if there are less than n observations in a node stop splitting; (ii) if a node is the n^{th} layer in the tree stop splitting; and (iii) if $s\%$ of the y_i in a node belong to one particular subset stop splitting. This sort of stopping rules is not frequently used when building a classification tree.

A second type of stopping rules is misclassification rate dependent. Such stopping rules are functions on the probability of misclassifying. Considered are two stopping rules related to the probability of misclassifying, namely *goodness of split* and the *minimum cost rule*. We will use the definition given by Breiman et al. for the impurity function and the impurity measure that we need to define goodness of split:

"An impurity function is a function ϕ defined on the set of all J-tuples of numbers $(p_1, \ldots, p_J)$ satisfying $p_j \geq 0, j = 1, \ldots, J, \Sigma_j pj = 1$ with the properties

(i) ϕ is a maximum only at the point $\left(\frac{1}{j}, \frac{1}{j}, \ldots, \frac{1}{j}\right)$,

(ii) ϕ achieves its minimum only at the points
$(1, 0, \ldots, 0), (0, 1, 0, \ldots, 0), \ldots, (0, 0, \ldots, 0, 1)$,

(iii) ϕ is a symmetric function of $p_1, \ldots, p_j$.

Given an impurity function ϕ, define the impurity measure $i(t)$ of any node t as

$$i(t) = \phi(p(1|t), p(2|t), \ldots, p(J|t)).$$

If a split s of a node t sends a proportion p_R of the data cases in t to t_R and the proportion p_L to t_L, define the decrease in impurity to be

$$\Delta i(s, t) = i(t) - p_R i(t_R) - p_L i(t_L).$$

Then take the goodness of split $\phi(s, t)$ to be $\Delta i(s, t)$."

In words the goodness of split is a value that reflects the improvement of classifying by splitting a set of data into two subsets compared to not splitting the set. In the literature two goodness of split stopping rules are often used: the *Gini index of diversity* and the *twoing rule*.

The Gini index of diversity is defined as:

$$i(t) = \sum_{i \neq j} p(i \mid t) p(j \mid t) .$$

With the twoing rule at a node t, with *s* splitting t into t_L and t_R, the split *s* that maximizes

$$\frac{p_L p_R}{4} \left[\sum_j | p(j \mid t_L) - p(j \mid t_R) | \right]^2 \qquad \text{is chosen.}$$

One can imagine that not in all cases a reasonable assumption is that misclassifying an outcome is of equal weight for all variables. Moreover, one could argue that misclassifying an *y* as a class *j*, that actually belongs to class *i*, is less desirable than misclassifying an *y*, that actually belongs to class *j*, as a class *i*. This is called the non-symmetry of classification costs. For example, assume *y* to be the outcome of an investment strategy with three classes (i) long, (ii) hold, and (iii) short; the costs associated with classifying a 'long' event as 'hold' event could be higher than the costs for classifying a 'hold' event as 'long' event, because in the first case investment opportunities are missed (high opportunity costs) and in the second case the costs of going long when there are no (negative) shocks are only the transaction costs (in the next period the position could be reduced if necessary).

The different misclassification costs can be represented by a set *C* with elements $C(i \mid j)$ for every class *i* and *j* satisfying (i) $C(i \mid j) \geq 0$, with $i \neq j$; and (ii) $C(i \mid j) = 0$ if $i = j$. Then for every node t the misclassification costs should be minimized and are defined as:

$$\sum_j C(i \mid j) p(j \mid t) .$$

The minimization of the classification costs is called the *minimum cost rule.*

Stopping rules are applied such that splitting is stopped in cases like: if $\max_{s \in S} \Delta i(s,t) < \beta$, for some threshold value β, then stop splitting (when using goodness of split measures); and if the minimum of the misclassification costs is greater than some threshold value β stop splitting. The use of such stopping rules has some disadvantages, which we will discuss now.

§ 2.2.4. Pruning

Classification trees built using one of the stopping rules can give misleading misclassification rates. If a very small threshold β is chosen, splitting will in most cases only stop if a node is pure (with a corresponding classification error of zero). There is a high probability that all terminal nodes consist of only one or two events. Based on the classification error one might think that a great job is being done, however if most terminal nodes consist one or two events the tree has no predictive power at all. Because all events are specified so precise, there is no reason to assume that a new event that follows the same path through the tree, ending in a terminal node with just one event from the past, will belong to that one particular class now or in the future. The only result we derived from the tree, is that we have split up the data set so far that every event can be identified as a unique one, clearly a case of data overfitting. The strengths of classification trees are undone, i.e. *no non-linear relationships* are revealed.

The use of more complicated stopping rules to overcome the problem of overfitting was investigated by Breiman et al.: "Using more complicated stopping rules did not help. Depending on the thresholding, the splitting was either stopped too soon at some terminal nodes or continued too far in other parts of the tree." The authors give a clear description on how and what kind of solution they found: "A satisfactory resolution came only after a fundamental shift in focus. Instead of attempting to the splitting at the right set of terminal nodes, continue the splitting until all terminal nodes are very small, resulting in a large tree. Selectively prune (recombine) this large tree upward, getting a decreasing sequence of subtrees. Then use cross-validation or test sample estimates to pick out that subtree having the lowest estimated misclassification tree ($R^*(d)$)."

A selection method often used to prune is *minimal cost-complex pruning*. The general idea of this method is that the cost-complexity measure is a function depending on both the misclassification rate and the complexity of the tree. This is comparable with the estimated R^2 in linear regression, i.e. whether or not to add a node or variable in the tree or regression based on the additive explanatory power of that particular node or variable. Define the complexity of any tree as $|\widetilde{T}|$ which is the number of terminal nodes. Let α be a non-negative real number that we call the *complexity parameter*. The *cost complexity measure* $R_\alpha(T)$ can now be defined as $R_\alpha(T) = R(T) + \alpha |\widetilde{T}|$. Interpret the cost-complexity parameter as a linear combination of the accuracy and the complexity of the tree. α can be interpreted as the complexity cost for every extra node in the tree, so if the explanatory power of adding an extra node does not outweigh the cost for adding it, the node will be cut off in the tree. Finding the optimal tree with minimal cost-complex pruning now becomes a minimization problem: $R_\alpha(T(\alpha)) = \min_{T \le T_{\max}} R_\alpha(T)$, with $T_{\max}$ the tree with the maximum number of nodes (so all nodes are pure consisting of one or more cases).

The algorithm for minimal cost-complex pruning is given by Breiman et al.: "Starting with T_1, the smallest subtree of T_{max} satisfying $R(T_1) = R(T_{max})$, the heart of minimal cost-complexity lies in understanding that it works by *weakest-link cutting*. For any node $t \in T_1$, denote by $\{t\}$ the subbranch of T_t consisting of the single node $\{t\}$.

Set $R_\alpha(\{t\}) = R(T) + \alpha$.

For any branch T_t, define

$$R_\alpha(T_t) = R(T_t) + \alpha |\widetilde{T}_t|.$$

As long as

$$R_\alpha(T_t) < R_\alpha(\{t\}),$$

the branch T_t has a smaller cost-complexity than the single node $\{t\}$. But at some critical value of α, the two cost-complexities become equal. At this point the subbranch $\{t\}$ is smaller than T_t, has the same cost-complexity, and is therefore preferable. To find this critical value of α, solve the inequality

$$R_\alpha(T_t) < R_\alpha(\{t\}),$$

getting

$$\alpha < \frac{R(t) - R(T_t)}{|\widetilde{T}_t| - 1}$$

By the assumption of $R(t) > R(T_t)$, the critical value on the right of above inequality is positive. Define a function $g_1(t)$, $t \in T_1$, by

$$g_1(t) = \begin{cases} \dfrac{R(t) - R(T_t)}{|\widetilde{T}_t| - 1}, t \notin \widetilde{T}_1 \\ +\infty, t \in \widetilde{T}_1 \end{cases}$$

Then define the *weakest link* $\bar{t}_1$ *in* T_1 as the node such that

$$g_1(\bar{t}_1) = \min_{t \in T_1} g_1(t)$$

and put

$$\alpha_2 = g_1(\bar{t}_1).$$

The node $\bar{t}_1$ is the weakest link in the sense that as the parameter α increases, it is the first node such that $R_\alpha(\{t\})$ becomes equal to $R_\alpha(T_t)$. Then $\{\bar{t}_1\}$ becomes preferable to $T_{\bar{t}_1}$, and α_2 is the value of α at which equality occurs.

Define a new tree $T_2 \prec T_1 - T_{\bar{t}_1}$.

Now, using T_2 instead of T_1, *find the weakest link in* T_2."

Important is that if there is a multiplicity of weakest links at any stage of the algorithm, i.e. if $g_k(\bar{t}_k) = g_k(\bar{t}_k^*)$, then define $T_{k+1} = T_k - T_{\bar{t}_k} - T_{\bar{t}_k^*}$. Repeat the minimal cost-complex pruning algorithm until the tree only consists of only the initial node (thus the set X as a whole). Note that this implies that the number of steps in the algorithm is smaller or equal to the number of nodes in T_{max}.

Weakest-link cutting results in a decreasing sequence of subtrees $T_1 \succ T_2 \succ \ldots \succ \{t_1\}$, where $T_k = T(a_k)$, $a_1 = 0$. The next step is to select the *optimum-sized tree*. The procedure proposed by Breiman et al. (1984) to find an optimum-sized tree is relatively easy. Their procedure, called 1-SE rule, is also used in some of the other classification tree algorithms we discuss in the next section. The 1-SE rule consists of four steps:

1.) Determine the misclassification rate $R(T_k)$ for each subtree, found by weakest-link cutting, based on the existing learning sample. Obviously $R(T_k)$ increases as $|\widetilde{T}_k|$ decreases.
2.) Determine the misclassification rate based on V-fold cross validation, i.e. $R^{cv}(T_k)$, or based on a test sample, i.e. $R^{ts}(T_k)$. Both methods are discussed in §2.1.4. Remark that $R^{cv}(T_k)$ or $R^{ts}(T_k)$ does not necessarily increase as $|\widetilde{T}_k|$ decreases!
3.) Determine the standard error for either $R^{cv}(T_k)$ or $R^{ts}(T_k)$. The standard error is defined as $\sqrt{\dfrac{R_{\min}(1 - R_{\min})}{\#observations}}$.
4.) Select the smallest tree for which the misclassification rate found in step 2 is within one standard error of the misclassification rate found in step 1 (that is why this rule is called the 1-SE rule).

In the literature the 1-SE rule is sometimes debated, because it should *overprune* the classification tree. A popular alternative is to choose in step 4 for the smallest absolute error, called 0-SE rule. Nevertheless, it is generally agreed that the 'best' of the two prune rules heavily depends on the involved data set. Therefore we will investigate which of two pruning rules is the 'best' for our data set in the empirical part of this thesis.

§ 2.2.5. The assignment of each terminal node to a class

The last step in the tree building procedure, the assignment of each terminal node to a class, is rather straightforward. That is, define class *i* to a node *t* if:

$$\sum_{i=1}^{J} C(i \mid j) p(j \mid t) \leq \min\left\{ \sum_{j=1}^{J} C(m \mid j) p(j \mid t) : 1 \leq m \leq J \right\}$$

In other words, node *t* is assigned to a class *i*, *given the selected optimum-sized tree* in case of minimal misclassification costs.

§ 2.2.6. Advantages of the tree structured approach

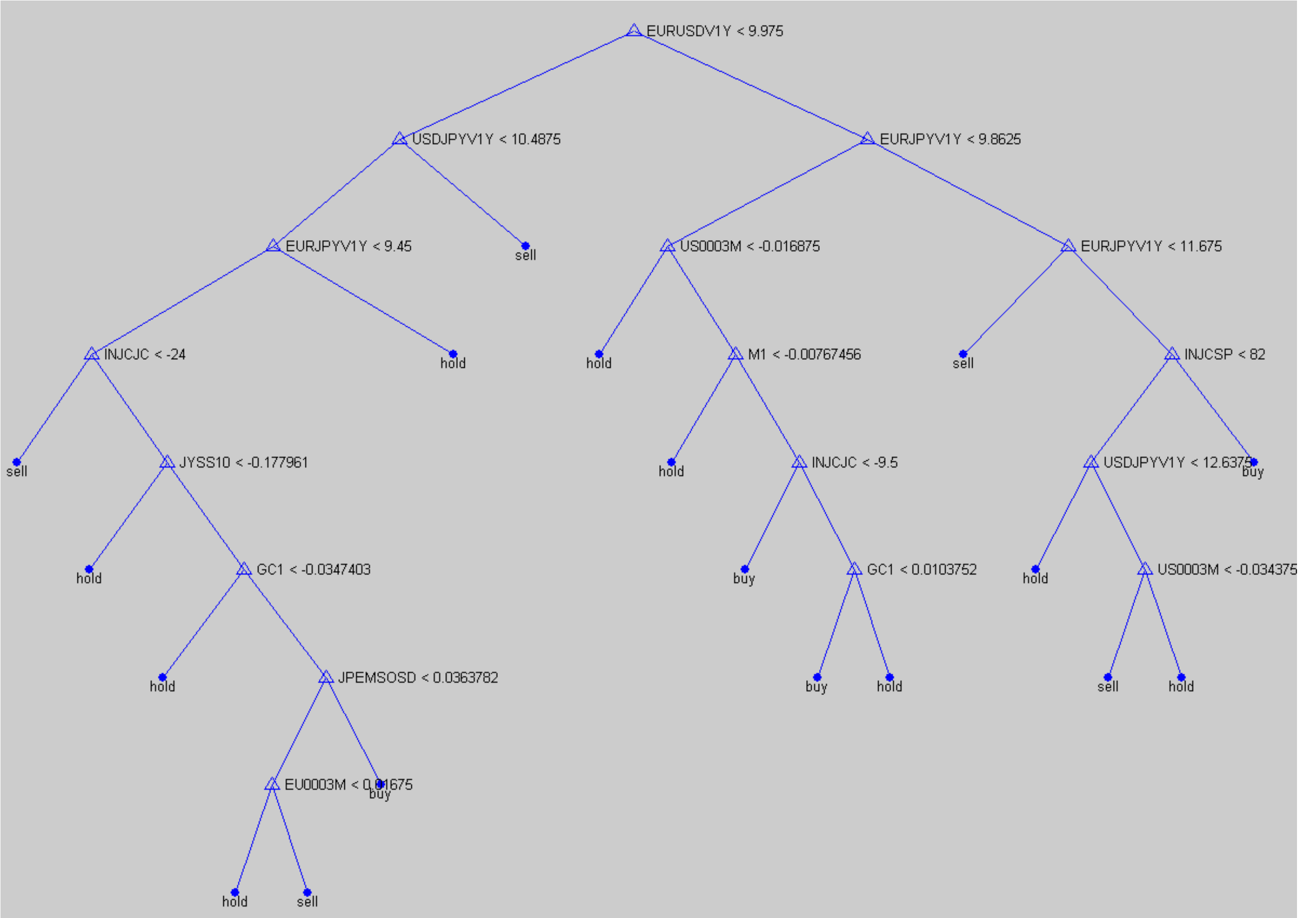
In the next section different classification trees will be discussed together with advantages and disadvantages of each particular algorithm. Nevertheless, classifying data by the use of a tree structured approach has some major advantages in general which we like to point out before the discussion about which algorithm to use is started. Classification trees can be powerful and flexible data analyser tools due to these properties:

1.) Independent of the data structure classification trees can deal with both ordered and categorical variables.
2.) If a tree is formed it can easily be stored and used to classify new data.
3.) It makes use of conditional information to handle and find nonhomogeneous relationships in a proper way.
4.) Both classification and misclassification rates can be estimated.
5.) It is invariant under all monotone transformations of individual ordered variables.
6.) It is very robust with respect to outliers and misclassification points in the learning sample.
7.) The way data is structured in a tree is easy to understand, the results are also easy to interpret.

§ 2.2.7. Graphical example of a classification

To illustrate how a completely classification tree looks we give a graphical example. In figure 1 (next page) a binomial tree is represented. To each node a variable name is added[1] together with the corresponding variable split value. At the end of tree path we find the terminal nodes. To each terminal node a (investment decision) class is assigned.

[1] For the specification of the variable see appendix C



Graph 1. An example of a (binomial) classification tree

Section 3 Classification tree algorithms

In this third section of the theoretical framework five different classification tree algorithms are discussed. Two of these algorithms are used to construct binary trees, the other three algorithms to construct mulitway trees. In theory every mulitway tree can be rebuilt by a binary tree, since every split in a multiway tree can be replicated by a sequence of binary splits. Still we discuss mulitway tree building algorithms, because a mulitway classification tree (could) have an advantage over a binary tree. Namely, by splitting each subset of the learning sample into two disjoint subsets the possibility exists that binary trees end up with terminal nodes (if the misclassification rate is allowed to be large enough and the x_i's of a certain class j are dispersed enough) for which some class j is not assigned to one of the terminal nodes. This problem *can in some cases* be avoided by the use of multiway splits. Therefore it is also very interesting to consider multiway classification trees.

§ 2.3.1. CART

CART stands for Classification And Regression Trees and is the general name for the classifiers developed by Breiman, Friedman, Olshen, and Stone. (1984). We only take the classification tree algorithm of Breiman et al. into account and refer to it as CART. Although the authors of CART title THAID as "the ancestor classification programme", in all the literature since 1984 about classification trees the book 'Classification And Regression Trees' of Breiman et al. is always used as the starting point.

The CART classification algorithm first determines for each variable the optimal split point within a subset. Secondly, it identifies the 'best' split variable to divide the subset into two smaller disjoint subsets. Such a sequence for the split of a subset is called *exhaustive search*. In CART two splitting rules are suggested: (i) the Gini criterion; and (ii) the twoing criterion. In the previous section we referred to these criteria as stopping rules, but the authors of CART prefer pruning methods over the use of stopping rules. The reason why both criteria can be used in CART as splitting rules is because they have some desirable mathematical properties that we will discuss next.

Focusing on just reducing the misclassification error while building a classification tree is optimal for every split, but not necessarily optimal for the tree as a whole. By minimizing classification errors stepwise, suboptimal splits are in general to be found due to the linearity in misclassification rates. This means that the split of a subset into one 'pure node' and a 'rest subset', which in consecutive splits can be divided into other pure nodes, is not preferred over the split of a subset into two 'less purer nodes' when the decrease of the misclassification rate is the same in both cases; the result is suboptimal splitting. Therefore Breiman et al. argue that the use of impurity functions ϕ that are strictly concave is preferred. Such impurity functions will select 'purer nodes' over 'less purer nodes' and are for that reason more suited for building classification trees. In other words, when building a classification tree one wants to find splits such that subsets with the highest concentration of some class j are selected. The other observations will be put together in the disjoint subset and again in that node a variable splitting should be found that separates a class j. Note that the 'rest subset' can still contain observations from a class j that are separated from that subset. At that moment only some path in the tree, where class j occurs, has been found. Other paths in the tree are very likely to exist, especially if the number of classes J is small.

Of course impurity functions that are strictly concave should also be well adjusted for probability measures. Therefore such impurity functions are defined as the class F of functions $\phi(p_i)$, $0 \leq p_i \leq 1$, with continuous second derivatives on $0 \leq p_i \leq 1$ satisfying: (i) $\phi(0) = \phi(1) = 0$; (ii) $\phi(p_i) = \phi(1 - \sum_{i \neq j} p_j)$; (iii) $\phi''(p_i) < 0$, $0 < p_i < 1$. Breiman et al. proof that both the Gini criterion and the twoing criterion satisfy these conditions. They also study the performance of both criteria on different data sets to find out if one of them outperforms the other and they conclude: "Choice of a criterion depends on the problem and on what information is desired. The final classification rule generated seems to be quite insensitive to the choice." Remarkably enough the authors do not give any statement about which rule is the 'best' one to use for splitting for each type of data set.

Rather than using a stopping rule Breiman et al. prefer to use the principal of pruning to find a right sized tree. "However, the tentative conclusion we have reached is that within a wide range of splitting criteria the properties of the final tree selected are surprisingly insensitive to the choice of splitting rule. The criterion used to prune or recombine upward is much more important." The algorithm how to implement pruning in a classification tree building procedure is already given in §2.2.4. For our research we will both use the Gini criterion and the twoing rule in the CART algorithm and test if one of them is of better use for our purpose.

§ 2.3.2. QUEST

The Quick, Unbiased, Efficient, Statistical Tree, in short QUEST, is the binary classification tree building algorithm introduced by Loh and Shih (1997). This method is said to have no bias towards the selection of variables that afford more splits (CART does have this bias), because it makes use of a type of discriminant analysis (will be discussed in §2.3.4.). Furthermore, the use of discriminant analysis makes it a much faster algorithm than those that make use of exhaustive search.

The QUEST algorithm first selects the variable that will be split and then finds the split point of that variable. At first for all *ordered* variables the levels of statistical significance are calculated by the use of the ANOVA F-statistic[2]. Secondly, for all *categorical* variables a Pearson contingency table X^2-test of independence[1] between the class variable and the categorical variable is used. The statistical significance of a categorical variable can easily be approximated with the chi-square distribution with $(J_t - 1)(M_t - 1)$ degrees of freedom, where M_t is the number of distinct categories present in the learning sample in node t. Based on the F-statistic (ordered variables) or the X^2-test (categorized variables) a P-value[1] can be calculated (stage I). "If the smallest P-value is less than a predefined threshold (determined via the Bonferroni method for multiple comparisons[*]), the corresponding variable is selected. Otherwise, Levene's F-test for unequal variances[1] is computed for each ordered variables (stage II). If the smallest P-value from the stage II-tests is less than another Bonferroni threshold, the corresponding variable is selected. Otherwise, the variable with the smallest P-value from stage I is selected." (Loh and Shih, 1997).

[2] A description of this statistical test or a definition of this statistic can be found in Appendix A

After the variable split selection for a node, the subset of that node is divided into two superclasses before application of the discriminant analysis. Obviously it is possible that those two superclasses have unequal variances. To cope with this problem, a modified version of quadratic discriminant analysis (QDA) is used. Because ordered variables are different than categorical variables, for each type of variable a different split point selection algorithm is used in the QUEST method.

QUEST consists of three algorithms: an algorithm for variable selection (i.e. algorithm 1. Variable selection in QUEST[3]); an algorithm for the split selection in case of an ordered variable (i.e. algorithm 2. Split selection for an ordered variable in QUEST[2]); and an algorithm for the split selection in case of a categorical variable (i.e. algorithm 3. Split selection for a categorical variable in QUEST[2]). In QUEST the classification can be constructed with the use of a stopping rule or by pruning as introduced by Breiman et al..

§ 2.3.3 CHAID

CHAID is the technique to classify data and is developed by Kass (1980). CHAID is the abbreviation of CHi-squared Automatic Interaction Detection and is mainly based on AID, which was introduced by Morgan and Sonquist (1963). The classification algorithm CHAID has some adjustments compared to AID that makes it the first classification algorithm that was widely used. The general idea of CHAID is that it selects the 'most significant' split (in the split selection) rather than the 'most explanatory'; *only categorized* predictors are considered. By focusing on selecting the 'most explanatory' (as being done in AID) the number of categories of the predictor is taken into account; the AID algorithm is biased in selecting predictors with more categories to be the split variable, because the maximization criterion used in this method extends over more possibilities. CHAID was developed to undo that bias.

In the selection process the best split for each predictor is found first. Then they are compared to each other and the best one is chosen. The procedure works like this: per predictor all combinations of the categories[4] are made and a significance level according to the chi-squared test is done. The chi-squared test relates the combinations of the predictor categories to the categories of the dependent variable with J-1 degrees of freedom. The combination with the highest significance level is selected as the 'best split' for that particular predictor. By looking per predictor at all the combinations of categories CHAID is a classification algorithms that allow multiway splits, for an example of possibilities in a multiway split see footnote 1.

We will we not discuss the CHAID algorithm here in any more detail, because our financial data is not categorical (at least not in a regular way) and therefore this classification algorithm will not be tested. Nevertheless, later developed multiway classification algorithms that we will use for our empirical research are based on the ideas of CHAID.

§ 2.3.4. FACT

Loh and Vanichsetakul (1988) present a method for classification called FACT which stands for Fast Algortihm for Classification Trees. The main characteristics of FACT are: (i) multiway splits are possible, (ii) uses a direct stopping rule, (iii) is not invariant of monotone transformations of

---

[3] These algorithms are given in Appendix B

[4] Example: assume a predictor containing the categories 1,2, and 3. Possible combinations are: 1/2/3, 1&2/3, 1&3/2, and 2&3/1.

the individual variables, and (iv) is computationally faster than algorithms that make use of exhaustive search.

When using FACT the problems of variable selection and split selection are separated. At each node the *F*-statistic in the ANOVA-table is calculated for every ordered variable. We select the variable with the largest *F*-statistic as the variable that will be split at that node. The split point selection for that variable is based on the technique Linear Discriminant Analysis (LDA). By the use of LDA it is assumed that by splitting a variable into two disjoint subsets the probability density functions $p(x \mid y = 0)$ and $p(x \mid y = 1)$ are both normally distributed. In other words: assume that *y* consists of four classes; the right splitting point is the one where we can make two subsets of *y* such that the variable *x* is normally distributed in both subsets (e.g. classes 1&3 and classes 2&4 could be combined for *y*, or classes 1&2&4 could be separated from class 3). Under this assumption, "a split is selected at node *t* via the discriminant function:

$$d_j(y) = \hat{\mu}_j{}'\hat{\Sigma}^{-1}y - \tfrac{1}{2}\hat{\mu}_j{}'\hat{\Sigma}^{-1}\hat{\mu}_j + \ln\{p(j \mid t)\},$$

where *y* denotes a vector in the space of the larger principal components, $\hat{\mu}_j$ is the sample mean vector of the *j*th class, and $\hat{\Sigma}$ is the sample pooled estimate of the covariance matrix at the node. Each node is split into *J* subnodes, and an object is channelled into the *i*th subnode if the latter minimizes the estimated expected misclassification cost:

$$\sum_{j=1}^{J} C(i \mid j)\exp\{d_j(y)\} = \min_{1 \le m \le J} \sum_{j=1}^{J} C(m \mid j)\exp\{d_j(y)\}.$$" (Loh and Vanichsetakul, 1988)

To deal with categorical variables in the FACT method they are transformed into ordered variables according to the so called CRIMCOORD method[5] which we will not discuss here (criticism on this transformation method is given by Breiman and Friedman, 1988).

In the FACT algorithm a direct stopping rule is used. Splitting is stopped if either the node error rate does not decrease, or if there is *at most* one class in the node with a sample size larger or equal to a by the user pre-defined size. The assignment rule is the same as in CART.

§ 2.3.5. CRUISE

The last (mulitway) classification tree building algorithm we discuss is the CRUISE algorithm introduced by Kim and Loh (2001). CRUISE is short for Classification Rule with Unbiased Interaction Selection and Estimation. The authors argue that this multiway classification algorithm has almost no variable selection bias (just like the QUEST algorithm). Due to the unbiasedness of variable selection in the CRUISE algorithm it is preferred over CHAID and FACT.

QUEST first selects the variable that will be split and then selects the split point of that variable. For the variable split selection Kim and Loh (2001) present two methods. The first is the same as the one used in FACT. The disadvantage of that particular variable split selection algorithm is that it is designed for testing the statistical significance of variables with unequal mean and variance, but not for testing statistical significance of variables with equal mean and variance *but with different distributions*. Therefore a second variable split selection algorithm is suggested. "The idea is to divide the space spanned by a pair of variables into regions and cross-tabulate the data using the regions as columns and the class labels as rows. The Pearson chi-

[5] Algorithm 4. CRIMCOORD transformation for categorical variables, can be found in Appendix B

square test of independence will be used to test its statistical significance. If both X_1 and X_2 are categorical variables, their category value pairs may be used to form the columns. If one variable is numerical and the other categorical, the former can be converted into a two-category variable by grouping its values according to whether they are larger or smaller than their sample median. To detect marginal effects we apply the same idea to each variable singly. If the variable is categorical, its categories form the columns of the table. If the variable is numerical, the columns are formed by dividing the values at the sample quartiles. Thus a set of marginal tables and a set of pairwise tables are obtained. The table with the most significant *p*-value is selected. If it is a marginal table, the associated variable is selected to split the node. Otherwise, if it is a pairwise table, we can choose the variable in the pair that has the smaller marginal *p*-value."

The two splitting algorithms for CRUISE given by Kim and Loh (2001) can be found in Appendix B (they have the names: Algorithm 5. Splitting algorithm 1D in CRUISE and Algorithm 6. Splitting algorithm 2D in CRUISE). When the learning sample contains no categorical variable splitting algorithm 1D is the same as the splitting algorithm used in FACT (as can be found in the appendix). This will also be the case in our empirical research later on. Therefore only the 2D splitting algorithm is used for CRUISE in the remainder of this thesis.

In the CRUISE algorithm linear discriminant analysis is also used to select the variable split point. Because LDA best works on data which are normally distributed and there is no reason to assume our data has such a distribution, first a Box-Cox transformation has to be performed. Box-Cox transformations are used to transform the distribution of data to a normal distribution. After a split point is selected the data will be transformed back to their original scale. In case of using the FACT split selection algorithm, no Box-Cox transformation (Algorithm 8. Box-Cox transformation in Appendix B) has to be performed for variables selected by Levene's test. In the CRUISE algorithm one can again choose to use of the known stopping rules.

Chapter 3. The empirical framework

Section 1. Degrees of freedom in model settings

In the previous chapter we discussed the CART with Gini Index, CART with Twoing rule, QUEST, FACT, and CRUISE classification algorithms. All these five algorithms are taken into consideration to determine which one is a good combination of powerfulness and reliability to predict the 10-Year Bund interest rate movement on a short time horizon. With a powerful algorithm we mean the one with the highest correct prediction rate of the interest rate movement; with a reliable algorithm we mean the one that has robust (statistical) performance when small changes in the model settings are made. The use of a classification algorithm brings along the decisions about quite a number of degrees of freedom in the model settings. Because we do not want to exclude some possible valuable model settings (i.e. choice of degrees of freedom) we approach our problem from a very broad perspective. The degrees of freedom we will have a closer look at are:

- Type of input database to use as set of possible explanatory variables
- The time horizon for which to predict the interest rate movement
- Number of possible classes to classify the interest rate movement
- The split minimum to use in the classification algorithms

§ 3.1.1.Type of input database

In principle every variable can be added to an input database for classification algorithms, because the different variable selection procedures are such that they *should* select the most explanatory variables at each variable selection step in an algorithm. On the other hand, a problem of classification algorithms is the decrease in selection power of the algorithms when a learning sample contains a large number of variables. The larger the dataset, the greater the possibility that the algorithms find a relation between one of the variables and the classification of the objective while in fact this relation is just random coincidence. We call this overfitting of the data set. Furthermore we favor a model that selects variables that make sense from an economic perspective. At last, we have to think about whether absolute levels of variables or its difference compared to a previous observation has the most possible explanatory power. Therefore we choose to consider the following three databases:

- A database with 81 economic variables from the US and Euro area, 17 of the variables are available at a weekly frequency and 64 at a monthly frequency; we look at the difference compared to the last known observation, in case of the monthly data we have three times a zero and then the monthly difference. Of course the differences are corrected for trends if necessary.
- A database with the (absolute or relative) differences of the 17 economic variables available at a weekly frequency as used in the first input database.
- A combination of the differences of the 17 economic variables available at a weekly frequency and 14 risk indicators (like volatility levels) as absolute level or as difference, also available at a weekly frequency.

All variables and risk indicators are available from January 1999 until October 2007, in total we use 455 weeks/observations as learning sample in the classification algorithms.

§ 3.1.2. The time horizon

For investment purposes we are only interested in the movement of the 10-Year Bund interest rate movement over a short time horizon. But because we do not know the time lag of the influence of economic data and risk indicators on that movement – possibly we find other explanatory variables and risk indicators for different time horizons – we choose to consider the following three time horizons:

- 1 week ahead
- 2 weeks ahead
- 4 weeks ahead

§ 3.1.3. Number of classes to classify interest rate movements

The main objective of our research is to predict an increasing or decreasing interest rate, so investment bets on the interest rate movements can be made. One could argue that it is therefore sufficient to classify the interest rate movement in only two classes, an 'up' and a 'down' class. But from an investor perspective an increase or decrease of the interest rate with only a few base points in a number of weeks is not very interesting. In fact, classification algorithms will not be very successful in classifying interest rate movements correctly with such a classification. Small deviations of the interest rate movement are most likely not (strongly) dependent on the economic variables which are used as input, possibly other non-data market factors do. Also the use of the multiway classification algorithms FACT and CRUISE are not that interesting anymore in a two-class classification problem. One could even argue that changes of some economic variables only influence large movements of the interest rate, whereas other market factors cause the relative smaller changes. To conclude, we have chosen to consider the following class combinations of the interest rate movement:

- 3 classes: a 'sell' class (increase in interest rate is expected), a 'hold' class, and a 'buy' class (decrease in interest rate is expected).
- 4 classes: an 'strong sell' class (interest rate is expected to raise heavily), an 'sell' class (small interest rate raise) and the opposite 'buy' class and 'strong sell' class.
- 5 classes: the same as 4 classes only in the middle a 'hold' class is incorporated to which only very small interest rate movements to both sides belong.

We have predefined the corresponding interest rate movement to the different classes for the different time horizons on forehand. The class boundaries are symmetric and chosen such that they are easy to interpret and have a relative uniform distribution in the learning sample. An overview of the class levels and the number of observations for each class for all time horizons is given in table 1.

LEVELS		1 Week	2 Weeks	4 Weeks
3 classes	*buy*	< -5bp	< -5bp	< -10bp
	hold	< +5bp	< +5bp	< +10bp
	sell	> +5bp	> +5bp	> +10bp
4 classes	*strong buy*	< -7.5bp	< -10bp	< -15bp
	buy	< 0bp	< 0bp	< 0bp
	sell	< +7.5bp	< +10bp	< +15bp
	strong sell	> +7.5bp	> +10bp	> +15bp
5 classes	*strong buy*	< -10bp	< -15bp	< -15bp
	buy	< -3bp	< -5bp	< -5bp
	hold	< +3bp	< +5bp	< +5bp
	sell	< +10bp	< +15bp	< +15bp
	strong sell	> 10bp	> 15bp	> 15bp

OBSERVATIONS		1 Week	2 Weeks	4 Weeks
3 classes	*buy*	129	164	136
	hold	205	136	176
	sell	121	154	140
4 classes	*strong buy*	90	93	98
	buy	148	138	137
	sell	127	118	111
	strong sell	90	105	106
5 classes	*strong buy*	52	48	98
	buy	118	116	91
	hold	132	136	88
	sell	86	98	69
	strong sell	67	56	106

Table 1: Overview of class levels and the number of observations per class for all time horizons

In a further stage of our research we will change the boundaries of these classes to see whether portfolio performance for the selected model settings are robust and even as important whether the portfolio performance can be improved by such boundary changes.

§ 3.1.4. Split minimum

We want to compare the statistical performance of the different classification algorithms with each other. Therefore we decide not to include pruning, because in case of pruning different models will have different pruning levels. Instead we predefine the split minimum in the classification trees and we will focus on performance differences as result of the predefined split minimum. We are particularly interested if it is more useful to have splits in the tree very deep (small split minimum) or are smaller trees more efficient (larger split minimums) for our problem. A second reason to look at quite a number of split minimums is to examine if the statistical performance of a classification tree does not fluctuate unacceptably when the split minimum is changed just a little bit. We will look and compare the results of classification trees with:

- a split minimum of 10
- a split minimum of 15
- a split minimum of 20
- a split minimum of 25
- a split minimum of 30
- a split minimum of 40
- a split minimum of 50

Classification algorithm		CART_1	CART_2	QUEST	FACT	CRUISE	
Input database		Monthly		Weekly		Weekly&RI	
Time horizon		1 week		2 weeks		4 weeks	
Number of classes		3 classes		4 classes		5 classes	
Split minimum in tree	50	40	30	25	20	15	10

Table 2. Overview of all model setting possibilities

CART_1 = CART with Gini Index
CART_2 = CART with Twoing Rule
RI = Risk Indicators

Section 2. Statistical measures

In total we will consider: 5 algorithms * 3 input databases * 3 time horizons * 3 class divisions * 7 split minimums = 945 model settings. To reduce this number quickly and efficient before we look at the portfolio performance of some models we have to define some criteria on which we base our decision to continue with certain model settings. Therefore, different statistical performance measures are calculated by taking the average value of the statistics over ten samples of 10-fold crossvalidation. So we divide the input database and classify outcomes in ten randomly chosen disjoint subsets, for each subset the interest rate movement is predicted based on the classification tree estimated by the combination of the other nine subsets. Then the statistical averages over the ten subsets are taken. We repeat this process of 10-fold crossvalidation ten times to have more reliable outcomes.

§ 3.2.1.Statistical measures used

Error rate:	The percentage of forecasts that are classified incorrectly
Error rate sell:	The percentage of forecasts that is classified as 'sell'-event, but should have been classified as 'buy'-event
Error rate buy:	The percentage of forecasts that is classified as 'buy'-event, but should have been classified as 'sell'-event
Error rate strong sell:	The percentage of forecasts that is classified as 'strong sell'-event, but should have been classified as 'buy'-event or 'strong buy'-event *Only available for '4 classes case' and '5 classes case'*
Error rate strong buy:	The percentage of forecasts that is classified as 'strong buy'-event, but should have been classified as 'sell'-event or 'strong sell'-event *Only available for '4 classes case' and '5 classes case'*
Corr. rate sell:	The percentage of forecasts that are classified as 'sell'-event, which initially have been classified as 'sell'-event
Corr. rate buy:	The percentage of forecasts that are classified as 'buy'-event, which initially have been classified as 'buy'-event
Corr. rate strong sell:	The percentage of forecasts that are classified as 'strong sell'-event, which initially have been classified as 'sell'-event or 'strong sell'-event *Only available for '4 classes case' and '5 classes case'*
Corr. rate strong buy:	The percentage of forecasts that are classified as 'strong buy'-event, and which have been classified as 'buy'-event or 'strong buy'-event *Only available for '4 classes case' and '5 classes case'*
Class distribution abs.:	The class distribution of the forecasts in absolute numbers

§ 3.2.2. Statistical performance versus portfolio performance

Different reasons cause us to look at statistical measures for all the different model settings instead of looking directly at portfolio performance. Most important is that calculating statistical measures by taking 10 samples of 10-fold crossvalidation per model setting enables us to use the available learning sample in an efficient way. Taking 10 samples gives us confidence in the reliability of statistical measures, because we can ground our decisions for model selection on the average and corresponding standard deviation. We can observe directly whether the statistical measures are robust or change heavily when just one of the model settings is changed. It is true that this could also be observed if you look at portfolio performance, but then you would just have one measure to base your decisions on.

The distribution of the classification over the classes is valuable information with respect to model behaviour. If a classification model, with for example 4 classes, classifies all observations as just one or two classes there is no reason to use that model. Other models with fewer classes are then more suited to classify data. A model that assigns almost no observations to certain classes is not satisfactory and reliable to use for future portfolio management, because we assume a constant distribution of interest rate movements over time. Models that assign almost no observations to one or more classes are therefore not useful. So, it is very important that the classification trees are able to classify data quite close to its 'true' distribution. The distribution statistics are thus a useful criterion to select the most promising model settings to continue our research with. As rule of thumb we say that if not at least 10% of the observations is assigned to each class, a model is not useful, i.e. at least 45 observations (10% of 455) have to be assigned to each class.

A second reason why we do not directly focus on portfolio performance is, because measurement of a portfolio performance brings along all kinds of different decision problems. Namely, the portfolio performance might be heavily dependent on the selected period over which to measure portfolio performance, the re-estimation frequency of the classification trees, and whether or not to use rolling windows as learning sample. These degrees of freedom regarding portfolio performance will be tested in the next stage of our research if the number of model settings is reduced to a much smaller one.

Section 3. Eliminating model settings

The most important result found is that models with weekly economic variables and the risk indicators as the input database have overall by far the most satisfactory statistical performance[6]. Correct prediction ratios are in general the highest for algorithms using this database and the rule of thumb for the distribution of the classification is in most cases satisfied. As it has turned out the satisfaction of the rule of thumb is very important for multiway trees. Apparently the use of only (weekly available) economic variables is not sufficient enough for classification trees. Also from an economic perspective we think that the combination of (frequent available) economic data changes and the absolute levels of risk indicators like volatility are indeed a very good combination to classify interest rate movements. In case of the largest database, we think that especially the large numbers of zeros in the database made it impossible to find variables with distinguishing power. Especially the use of Linear Discriminant Analysis (LDA) in some of the variable selection steps of the classification algorithms is not suited to deal with large number of non-available observations.

If we only focus on the models with the weekly variables and risk indicators as input database the first classification algorithms we can eliminate are the QUEST algorithm and the CART with Twoing rule algorithm. The first because QUEST clearly underperforms both the CART algorithms when statistical measures are compared. The second is eliminated because it has mostly the same outcomes over all model settings as the CART with Gini Index algorithm, we prefer that one because it is more often used in practice. So, we will continue our research with just one binomial classification tree. This does not mean that the Twoing rule is useless for our problem.

The choice between one of the two multiway classification trees isn't that easy. Both algorithms have *in some model settings* a very satisfactory and promising performance, but they can differ for the same model settings. Therefore both classification tree algorithms will be considered in future research.

The next step is to reduce the number of time horizons. Statistics of models with just one week as time horizon are not very robust and correct ratios are clearly below those for models with a time horizon of two or four weeks. Apparently, the effect of new economic data and risk indicators is not directly reflected in the interest rate in one week, possible it takes more time for the new data to be absorbed in the interest rate (and may also depend on other data that become available at a later point of time, these are not incorporated in our models).

We are not able to eliminate one of our three initial choices of class structures. The effect of such a choice will hopefully be more revealed when we look at portfolio performance.

As hoped the statistical performance of the different model settings is quite robust over the different split minimums. We like this because the choice of split minimum has apparently not a large impact on the model performance and the results are thus more reliable. On the other hand we are not able to determine if we should have a low split minimum or a high split minimum. Because of the robust outcomes from now on we choose to consider only models with a split minimum of 10, 25, and 50.

[6] In Appendix D the statistical measure results for the FACT algorithm with economic week data and risk indicators can be found as example, other results are omitted to save space

Overall we have reduced the number of model settings to: 3 algorithms * 1 input database * 2 time horizons * 3 class divisions * 3 split minimums = 54 model settings. In the next chapter we will try to find satisfactory classification models based on portfolio performance and the robustness of that performance.

Classification algorithm		CART_1	CART_2	QUEST	FACT	CRUISE	
Input database		Monthly		Weekly		Weekly&RI	
Time horizon		1 week	2 weeks		4 weeks		
Number of classes		3 classes	4 classes		5 classes		
Split minimum in tree	50	40	30	25	20	15	10

Selected model setting for further research

Eliminated model setting for further research

Table 3. Overview of model settings left after elimating some model settings

CART_1 = CART with Gini Index
CART_2 = CART with Twoing Rule
RI = Risk Indicators

Section 4. Portfolio construction and portfolio measure

The next step in our research is to link forecasts of the 10-Year Bund interest rate movement by the use of classification trees to a portfolio strategy. Three possible investment signals about an interest rate movement can be given when we classify a new set of economic variables and risk indicators based on an estimated classification tree. Which class, i.e. investment action, in which model is related to a certain signal is given in the following table.

Classification signal	3 classes	4 classes	5 classes
Interest rate is expected to rise	sell	strong sell sell	strong sell sell
Interest rate is expected to not move much	hold	xxx	hold
Interest rate is expected to go down	buy	buy strong buy	buy strong buy

Table 4. Classification signals and corresponding class assignment/investment action

To translate the classification signal into an investment strategy we first have to decide in which type of financial product to invest in. We choose to take a position in the 10-Year Bund Future[7] for two or four weeks depending on our forecast horizon. The price of the 10-Year Bund moves in the opposite direction as the interest rate does. The reason is quite simple: Assume that the current 10-Year Bund interest rate is equal to the interest rate of an existing 10-Year Bund. If the 10-Year Bund interest rate rises, an investor holding the existing 10-Year Bund receives a lower return than if he would buy a new Bund with the higher interest rate. So the price of the older 10-Year Bund will go down. The opposite holds for a falling interest rate.

Because we take a position in a future we do not need to invest any money at the time of purchase, such a strategy is called a zero-investment strategy. We will take a position every week (or no position in case of a 'mid' class forecast) in the 10 Year Bund future and take the relative return of the position after 2 or 4 weeks, depending on the time horizon, to measure the performance of the strategy. Transaction costs and possible margin account obligations are ignored.

The returns we make on our investment strategy are called α-performance[8]. To compare different returns of the α-performance for investment strategies based on different classification models we will look at the Sharpe Ratio. The Sharpe Ratio is a measure of the excess return per unit risk in our investment strategies and is introduced by Sharpe (1966). If a constant risk-free rate is assumed the Sharpe Ratio is defined as:

$$SharpeRatio = \frac{E[R] - R_f}{\sqrt{\mathrm{var}[R]}} \tag{4.1}$$

[7] A bond future is a contractual obligation for the contract holder to purchase or sell a bond on a specified date at a predetermined price. A bond future can be bought in a future exchange market and the prices and dates are determined at the time the future is purchased.

[8] α-performance is a measure of performance on a risk-adjusted basis. Alpha takes the volatility (price risk) of a mutual fund and compares its risk-adjusted performance to a benchmark index. The excess return of the fund relative to the return of the benchmark index is a fund's alpha.

Formula (4.1) is the Sharpe Ratio for an investment portfolio with an initial investment sum for which the risk-free rate could always be earned. Because our strategy is zero-investment strategy we cannot earn the risk-free rate on any money and should therefore not be included in the Sharpe Ratio to measure the α-performance. So the Sharpe Ratio reduces to:

$$SharpeRatio = \frac{E[R]}{\sqrt{\mathrm{var}[R]}}\,. \tag{4.2}$$

We consider a Sharpe Ratio of 0.5 as a very good alpha-performance, implying that the classification model succeeds in forecasting the 10-Year Bund interest rate movement resulting in an expected positive return without excessive return volatility.

To test whether the Sharpe Ratios are significant different from 0 at the 95% level we have to determine the confidence interval (CI) of the Sharpe Ratios.
We assume that all weekly returns R_1, R_2, …, R_t, … R_T are i.i.d.. In our empirical research we have 326 weekly returns (i.e. 6.5 years x 52 weeks), so T = 326. Furthermore it holds that:

Estimate (weekly)	True value (weekly)	True value (yearly)
$\hat{\mu}_W = \frac{1}{T}\sum_{t=1}^{T} R_t$	$\mu_W = ER_t$	$\mu_Y = 52 \cdot \mu_W$
$\hat{\sigma}_W = \sqrt{\frac{1}{T}\sum_{t=1}^{T}\left(R_t - \hat{\mu}_W\right)^2}$	$\sigma_W = \sqrt{Var(R_t)}$	$\sigma_Y = \sqrt{52}\cdot\sigma_W$
$\hat{S}R_W = \frac{\hat{\mu}}{\hat{\sigma}}$	$SR_W = \frac{\mu}{\sigma}$	$SR_Y = \sqrt{52} \cdot SR_W$

To determine the confidence interval of the Sharpe Ratios we have to know the variance of the estimated Sharpe Ratio, which is equal to:

$$Var\left(\hat{S}R\right) \approx \frac{Var(\hat{\mu})}{\hat{\sigma}^2} = \frac{{}^{1}\!/_{T} \cdot Var(R_t)}{\sigma^2} = \frac{1}{T} \tag{4.3}$$

Therefore the 95% confidence interval of the weekly and yearly Sharpe Ratios are:

$$CI_W = \hat{S}R_W \pm 1.96 \cdot \sqrt{{}^{1}\!/_{T}} = \hat{S}R_W \pm 1.96 \cdot \sqrt{{}^{1}\!/_{326}} = \hat{S}R_W \pm 0.1086 \tag{4.4}$$

$$CI_Y = \sqrt{52} \cdot \left(\hat{S}R_W \pm 0.1086\right) = \hat{S}R_Y \pm 0.7828 \tag{4.5}$$

Thus an estimated (yearly) Sharpe Ratio should be larger than 0.7828 to be significant different from 0 at the 95% confidence interval. We will give the confidence intervals of the Sharpe Ratios in the remainder of this thesis in brackets after the Sharpe Ratio. This level of 0.7828 is larger than our desired 0.5 level, nevertheless we feel that a Sharpe Ratio above the 0.5 level is still reasonable to focus our attention on.

Chapter 4. Portfolio performances

In this chapter the portfolio performance of the classification models with some adjustments will be discussed. To compare the robustness of different portfolio performances we will start with the standard model settings in §4.1.1. The use of cost functions will be introduced in §4.1.2. and a comparison in performance between the use of a rolling window or expanding window can be found in §4.1.3. The choice of different class borders will be discussed in §4.1.4. As we will see throughout this chapter the FACT algorithm has the best portfolio performance for models with 3 classes. Testing whether we made a good classification model choice is done in the second section of this chapter by looking at portfolio performances over the period outside the test period.

Section 1. Portfolio performance robustness

§ 4.1.1. Standard models: the portfolio performances

As discussed in the previous chapter we choose CART with Gini Index as the binomial tree to link to a portfolio performance. Because we were not able to make a good choice for the multiway trees between the FACT and the CRUISE model, we decided to consider both with respect to portfolio performances. In the standard models equal costs for each misclassification are assumed. More practically, if a terminal node is reached, i.e. the number of observations in a node is less than the split minimum, its class label will automatically correspond to the class with the most observations in that node.

At first we have to determine over which period we will measure a portfolio performance and how often we will re-estimate a classification model. We set the time window over which we will measure the performance as from 2001 until the half of 2007. It means that for the first period the classification models will be estimated with the use of two years of information (the years 1999 and 2000). For the estimation of the models over the remaining periods we will use all available information up to that period, i.e. we use an expanding window as learning sample. The question how often to re-estimate the classification models is more difficult to answer. Therefore we decide to use three re-estimation frequencies, so we will consider the portfolio performances in case of yearly re-estimation, semi-annually re-estimation, and quarterly re-estimation. A robust portfolio performance over the different re-estimation frequencies will definitely strengthen our confidence in a classification model.

In Appendix E the yearly average return, yearly average standard deviation of the returns and the yearly Sharpe Ratios for the different chosen models can be found per classification algorithm. 95% confidence intervals of the Sharpe Ratios are given between brackets.

CART with Gini index: The portfolio performances for the CART algorithm are not that good at all. Most promising are the portfolio performances of models with 4 classes and 2 weeks time horizon. 7 out of 9 Sharpe Ratios are positive (minimum is -0.02) and they are very high for the portfolios with quarterly re-estimation frequency. Those Sharpe Ratios are 0.55 (CI: -0.23, 1.33), 0.89 (CI: 0.11, 1.67), and 1.03 (CI: 0.25, 1.81) for a split minimum of respectively 10, 25, and 50. Also interesting are the Sharpe Ratios for the model with 3 classes and 2 weeks time horizon, 8 out of 9 Sharpe Ratios are positive. The portfolio performance of the other model settings are mostly negative. Furthermore, it isn't possible to say anything on the development of Sharpe Ratios when split minimums are increased or when the re-estimation frequency is higher.

FACT: Remarkably enough we find completely different portfolio performances for the FACT algorithm as than found for the CART with Gini index algorithm. The portfolio performance of the models with 3 classes are really good for the FACT algorithm. Only 2 out of 18 models with 3 classes have a negative Sharpe Ratio (just -0.0004 and –0.0595), and 8 of those 18 have a Sharpe Ratio above 0.5 (of which 5 Sharpe Ratios are at the 95% confidence interval significant different from 0). The best portfolio performance for each model with 3 classes is found for the semi-annually re-estimation frequency. The performances of the models with more than 3 classes are not good. Although we find three Sharpe Ratios above the 0.5 level for those models, we believe that these are just lucky coincidences, because if we look at the same models with (small) changes in split minimum or re-estimation frequency, Sharpe Ratios change dramatically. Meaning that the results for models with more than 3 classes are not robust.

CRUISE: The portfolio performances for the models with this algorithm are the worst of the three algorithms. Also for this multiway tree algorithm we find only robust positive Sharpe Ratios for models with 3 classes. Nevertheless, these Sharpe Ratios are far below the levels as found for the FACT algorithm.

From the analysis of the standard models we cannot choose one of the algorithms to be the best or most promising one yet. A very interesting observation is that CART performs well for models with 4 classes whereas FACT and CRUISE only perform well for models with 3 classes. This could be explained by how the two types, binomial versus multiway, classification trees work. The division of a set by a split of a variable into 4 or 5 disjoint subsets is not successful in our case, but mulitway trees with 3 classes per node are relatively well performing. Most likely we loose too much valuable information if we split a set into too many subsets to label terminal nodes correctly. Graphically one could say that we get a short and broad tree in these cases, the end of the tree is reached too fast. Whereas this is probably not the case when there are only three classes. For the CART algorithm that has binomial variable splits, we argue that especially the 'choice' of more classes in terminal nodes is of great value. At each step of the algorithm a subset of the larger set is separated such that one class is typical for that variable split. The larger the choice in classes the more easily it leads to a smaller split sub sample with less 'noisy' observations, but with a higher purity of a certain class in that subset. A hypothetical example where we focus on the error rate of the right distribution:

	3 classes	4 classes
Distribution node	(40, 30, 30)	(35, 25, 20, 20)
split left distr.	(15, 15, 15)	(10, 15, 10, 15)
split right distr.	(25, 15, 15)	(25, 10, 10, 5)
# samples right	55	50
err right distr.	30/55 = 6/11	25/50 =1/2

Table 5. Classification example

The error rate in the right split decreases in case of 4 classes even while the number of samples is also smaller compared to the case of 3 classes. In the next paragraph we will see if the performances of models with different number of classes also hold when differences in classification costs are included.

§ 4.1.2. Misclassification Costs

Relaxing the assumption of equal misclassifications costs makes sense, especially in a financial context. From an investor's point of view it is even not true that each type of prediction error has the same effect on a portfolio performance. Assume the three class model, the three possible classes/investment signals are (i) 'buy', (ii) 'hold', and (iii) 'sell'. Let's think from the perspective of a risk averse investor, the investor wants to have some confidence that the chance (based on historical data) of taking a 'buy' position' while Bund prices will go down is relative small. On the other hand he does not care much if the Bund price does not change significantly. So, the costs of misclassifying a 'hold' event as 'buy' event' are much smaller than the costs of misclassifying a 'sell' event as 'buy' event. For example, assume we have a terminal node and we have to assign a class label to this node. Given 5 'buy' observations, 1 'hold' observations, and 4 'buy' observations in the standard model the class label 'buy' would be assigned to the terminal node, but a risk aversion investor would rather want to assign the class label 'hold' because he does not have enough confidence in the fact that he will make money with such a decisions. In fact it could be that the 'sell' shocks are much larger than the 'buy' shocks and assigning 'buy' to this node is then a loosing money strategy. Unequal misclassification costs could prevent this problem as we point out in the next example:

Possible misclassification matrix

3 classes	*sell*	*hold*	*buy*	*Numb. obs.*
sell	0	1	3	#5
hold	1	0	1	#1
buy	3	1	0	#4

	Misclassification costs
Costs assigning class label 1	5*0 + 1*1 + 4*3 = 13
Costs assigning class label 2	5*1 + 1*0 + 4*1 = 9
Costs assigning class label 3	5*3 + 1*1 + 4*0 = 16
Assign class label:	**label 2: 'hold' position**

Table 6. Misclassification example

With a 'hold' position as resulting class label, risks of opposite investment decisions are reduced; Sharpe Ratios give us the information whether the (possible) loss of returns are compensated by the reduction in volatility of the returns.

In the above sample misclassification costs are symmetric, i.e. misclassifying a 'hold' event as 'buy' event has the same costs as misclassifying a 'buy' event as 'hold' event. Also this does not have to be the case for an investor. Let's repeat the example from §2.2.3.: "The costs associated with classifying a 'buy' event as 'hold' event could be higher than the costs for classifying a 'hold' event as 'buy' event, because in the first case investment opportunities are missed (high opportunity costs) and in the second case the costs of going buy when there are no (negative) shocks are only the transaction costs." This is called asymmetry of misclassification costs.

We introduce two cost matrices to test whether our thoughts about misclassification make sense in our investor setting. The first cost matrix we propose has only 'symmetric non-equal misclassification costs' and the second is a cost matrix with 'non-symmetric non-equal misclassification costs'. Note that for the matrix with non-symmetric misclassification costs and 4 classes it makes no sense from an investor perspective to include non-symmetric misclassification costs and are thus not incorporated, because we only have the buy and sell signal. Instead we look at another set of non-equal misclassification costs.

Symmetric non-equal misclassification cost matrices

3 classes	*sell*	*hold*	*buy*
sell	0	1	3
hold	1	0	1
buy	3	1	0

4 classes	*upup*	*up*	*down*	*downdown*
upup	0	1	3	3
up	1	0	3	3
down	3	3	0	1
downdown	3	3	1	0

5 classes	*upup*	*up*	*mid*	*down*	*downdown*
upup	0	1	1	3	3
up	1	0	1	3	3
=	1	1	0	1	1
down	3	3	1	0	1
downdown	3	3	1	1	0

Non-symmetric non-equal misclassification cost matrices

3 classes	*sell*	*hold*	*buy*
sell	0	0,5	2
hold	1	0	1
buy	2	0,5	0

4 classes	*upup*	*up*	*down*	*downdown*
upup	0	0,5	1	2
up	0,5	0	1	2
down	2	1	0	0,5
downdown	2	1	0,5	0

5 classes	*upup*	*up*	*mid*	*down*	*downdown*
upup	0	0,5	1	2	2
up	0,5	0	1	2	2
=	2	1	0	1	2
down	2	2	1	0	0,5
downdown	2	2	1	0,5	0

Table 7. Suggested misclassification cost matrices

In Appendix E the yearly average returns, yearly average standard deviation of the returns and the yearly Sharpe Ratios for the different model choices can be found per classification algorithm for both misclassification matrices. The most important results are discussed per algorithm and per suggested misclassification cost matrix.

- *Symmetric non-equal misclassification cost matrices*

CART with Gini index: By using the cost matrix no specific improvements for the portfolio performances are found. The Sharpe Ratios of the most promising model (4 classes and 4 weeks time horizon) have not increased. On the other hand 5 out of 9 Sharpe Ratios did not change at all and the other 4 Sharpe Ratios have decreased minimally. This indicates that the CART algorithm is robust for changing the cost matrix from equal misclassification costs to a cost matrix with symmetric non-equal misclassification costs. Unfortunately, the Sharpe Ratios for the models with 3 classes, 2 weeks time horizon, and quarterly re-estimation frequency are all negative, instead of positive in the standard model.

FACT: No clear improvements of Sharpe Ratios are found for models with the FACT algorithm. Good with respect to model robustness is to see that the portfolio performances for models with 3 classes are still positive.

CRUISE: Although we observe much more Sharpe Ratios above the 0.5 level as we saw earlier (11 versus 2), we also observe a lot of decreasing Sharpe Ratios for different models. We do not see any structure in the improvements. Also the performances of the 3 class models, which had the highest Sharpe Ratios in the standard models, became worse.

- *Non-symmetric non-equal misclassification cost matrices*

CART with Gini index: The results have not improved at all. The Sharpe Ratios for the models with 4 classes and 4 weeks time horizon have become worse. Clearly, the use of the non-symmetric non-equal misclassification cost matrices is not very helpful to improve portfolio performances for models with the CART algorithm.

FACT: We find really great Sharpe Ratios for the models with 3 classes. If we look at the portfolios that are re-estimated at a semi-annually and quarterly frequency then we find that 10 out of 12 Sharpe Ratios are above the 0.5 level. If we also include the portfolios with yearly re-estimation, none of the Sharpe Ratios is negative (the smallest is 0.07). For models with more classes we do not observe robust improvements.
CRUISE: With exception of the models with 4 classes and yearly and semi-annually re-estimation frequency, all Sharpe Ratios have decreased. When using the non-symmetric non-equal misclassification cost matrices the CRUISE algorithm again performs the worst of the three algorithms.

At the end of this paragraph we can conclude that the introduction of both symmetric and non-symmetric non-equal misclassification costs in our investor setting did not lead to improvements in portfolio performances, with one exception. The exception is the use of the non-symmetric non-equal misclassification cost matrix for 3 classes when using the FACT algorithm. It had the result that none of the 18 Sharpe Ratios for the models with 3 classes, two time horizons, and three re-estimation frequencies were negative. In fact 9 of 18 were above the 0.5 level. That we did not find any other improvements by the introduction of both non-equal misclassification costs does not mean they could not be of great value in some (model) settings. Further research, beyond the scope of this thesis, should be done to find the direct influence of non-equal misclassification cost matrices on the performance of investment strategies. In that stage one could also link this to the expected, desired, or initial distribution of classes. Implying that the class distribution of the learning sample is not necessarily always in line with the future (desired) class distribution. Such inequalities could probably be resolved by the use of misclassification cost matrices.

At this stage of our research we decide to drop the CART with Gini index algorithm as a good model for 10-Year Bund investment purposes. Although we find some promising results for the models with 4 classes and 4 weeks time horizon, we observe that the Sharpe Ratios for the FACT algorithm for models with 3 classes are much higher and we believe that those portfolio performances are more robust. We also drop the CRUISE algorithm as a good model for Bund investment purposes. Even though it showed some satisfying results for models with 3 classes, the FACT algorithm outperformed (based on Sharpe Ratios) these results. So, to test whether our choice for the use of an expanding window for the learning sample is a correct one, only the FACT algorithm is used.

§ 4.1.3. Expanding window versus rolling window

So far we only considered portfolio performances of models estimated with a learning sample collected by an expanding window; for every re-estimation we use all available information up to that moment in time. From a statistical point of view it is correct to do so. Namely, when using the classification algorithms we assume that the input is independently and identical distributed. In other words, it should hold that the distribution of our data, i.e. economic variables and risk indicators, does not change over time. An expanding window covers more data over time and it is therefore likely that, from this perspective, the data better fits its real (non-observable) distribution, i.e. the more data points, the more we are close to a data distribution (law of large numbers).

On the other hand this statistical behaviour has not to be true in financial practice. Why should we assume independently and identical distributed data? It is quite possible that the distribution of economic data changes over time by for example change in volatility; also the clustering of data is a well-known economic phenomenon. For our problem we can test this by using a rolling window as learning sample. A rolling window means that we only use the past *n* observations as learning sample for the re-estimation of a classification tree. We take a period of 2 years to test portfolio performance of models re-estimated with a rolling window, so there are always 104 observations in our learning sample.

The results for the FACT algorithm have definitely not improved. If we focus on the models with three classes for 13 of the 18 Sharpe Ratios we find lower values then in the standard model. Also the number of Sharpe Ratios above the 0.5 level has decreased to just 4 (compared to 10 in the standard model). For the models with more than 3 classes we find the most positive Sharpe Ratios for models with yearly re-estimation, just one is negative. On the other hand we find the most negative Sharpe Ratios for the models with quarterly re-estimation, just 4 are positive. Remark that these last results are not that important as the outcomes for the models with 3 classes, because the performances of these classes are much lower.

We can conclude this paragraph by stating that 2 years of data is overall a too short horizon to use as learning sample. One could also test the use of other rolling window lengths, but we decide to skip these tests. Main reason is the relative small size of our data set for which we therefore favour the use of an expanding window.

§ 4.1.4. Change of class boundaries

In this paragraph we will test whether portfolio results for the FACT algorithm are robust or even can be improved when the class boundaries are changed. We have chosen to look only at models with 3 classes. Because the number of models is reduced to just to two, the model with 3 classes and 2 weeks horizons and the model with 3 classes and 4 weeks time horizon, it is also more easy to test the robustness of the portfolio performance with respect to the split minimum. Therefore we will include split minimums of 10, 15, 20, 25, 35, and 50.

So far all research on model performances is based on a rather classification of the interest rate movements (see § 3.1.3.). Class boundaries have been chosen such that opposite movements are symmetric and the level of movement is easy to interpret, e.g. 3 classes and 4 weeks time horizon: a 'sell' classification if the increase is larger than 10 base points, a 'hold' classification if the change is no larger than 10 base points, and a 'buy' classification if the decrease is larger than 10 base points. The predefined classification of interest rate movements could have influenced our research. We do not have any guarantee that we will have the similar portfolio results for models with a (slightly) different input classification of the interest rate movements. Therefore we will study the robustness of the portfolio performances with respect to changes in class boundaries. We suggest three types of class boundary settings to classify the interest rate movements (keeping the number of possible class movements the same, thus 3), still satisfying the symmetry requirement.

The three adjusted class boundary settings are chosen such that:

- the number of observations are almost equal over all classes
- the extreme movements have twice as much observations as the less extreme movements
- the extreme movements have half the observations as the less extreme movements

Important to remark is that class label assignment in the models is based on a cost matrix with equal misclassifications as in the standard models. Thus, the misclassification costs of extreme movements in the opposite direction are three times the misclassification costs of other classes. The use of the last two class boundary settings will definitely have an impact on the model performance, but we cannot say on beforehand whether we expect extreme movements to be more or less predicted. Consequently, the first suggested class boundary change is the best proposal to test the robustness of our model choice. The adjusted class boundary settings can be found in table 8 together with the number of observations in the total learning sample:

LEVELS				
		adj1	adj2	adj3
3 classes	*sell*	< -5.5bp	< -3bp	< -8bp
2 weeks	*hold*	< +5.5bp	< +3bp	< +8bp
	buy	> +5.5bp	> +3bp	> +8bp
		adj1	adj2	adj3
3 classes	*sell*	< -9bp	< -6bp	< -12bp
4 weeks	*hold*	< +9bp	< +6bp	< +12bp
	buy	> +9bp	> +6bp	> +12bp

OBSERVATIONS				
		adj1	adj2	adj3
3 classes	*sell*	158	192	129
2 weeks	*hold*	147	79	199
	buy	149	183	126
		adj1	adj2	adj3
3 classes	*sell*	147	177	122
4 weeks	*hold*	153	111	204
	buy	152	164	126

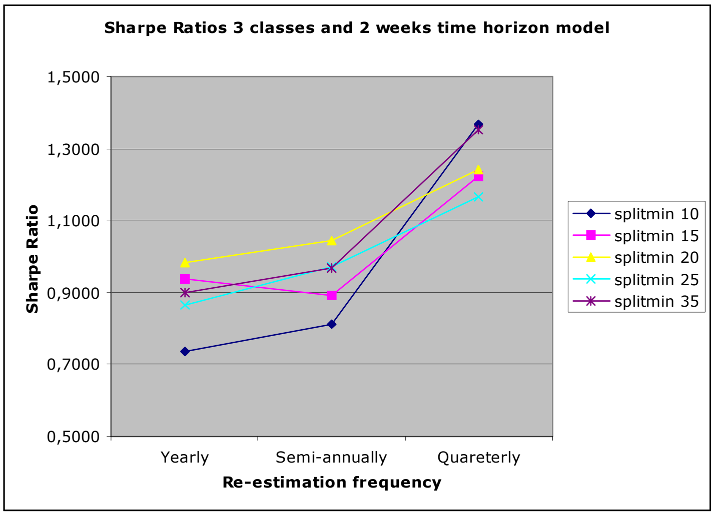
Table 8. Suggested class boundary settings

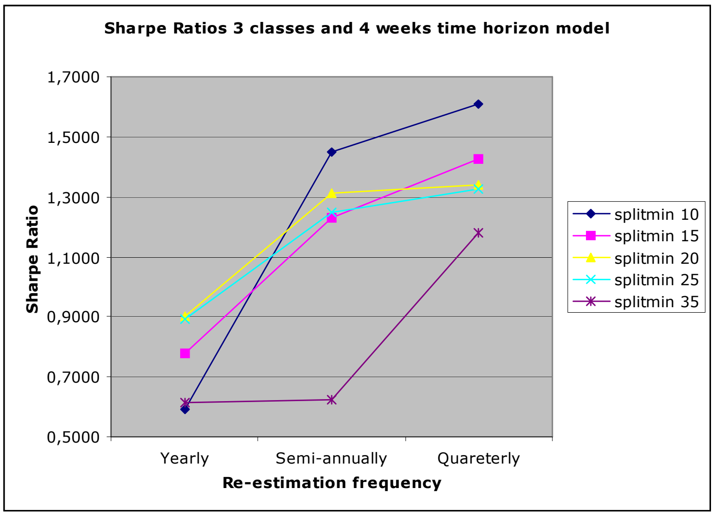
In Appendix E the yearly returns, yearly standard deviation of the returns and the yearly Sharpe Ratios can be found per adjusted class boundary setting for the two models we have selected up to now. The most important results are discussed.

- *Class boundary setting 1 (equal distribution of observations per class)*

  2 weeks time horizon: The results are superb. If we exclude the Sharpe Ratios of the models with a split minimum of 50 we observe only Sharpe Ratios above the 0.5 level. The minimum Sharpe Ratio is 0.74 (CI: -0.04, 1.52) and the maximum 1.37 (CI: 0.59, 2.15). Another important result to note is that (with exception of the model with a split minimum of 15) the Sharpe Ratio increases when the re-estimation frequency increases as shown in the graph 2. It means that with this choice of the class boundaries, including recent financial data in our model estimation leads to even improved forecasting ability of the model.

  4 weeks time horizon: The results for this model are even better (if we again exclude the split minimum of 50). All Sharpe Ratios are above the 0.5 level, with 0.59 (CI: -0.19, 1.37) the lowest and 1.60 (0.82, 2.38) as the highest. Remarkable is that these extremes are found for exact the same model settings as in the 2 weeks time horizon case. We find the minimums for the models with a split minimum of 10 and yearly re-estimation frequency, and we find the maximums for the models with also a split minimum of 10 but with quarterly re-estimation frequency. We think we can explain this as follows: if a model is very large (i.e. small split minimum) the end of the trees are very sensitive to just a small number of observations. Therefore it is important 'to keep the ends of the tree up-to-date'. When the trees are only re-estimated yearly, they are thus not really 'fine-tuned' with respect to recent developments in financial markets.





Graphs 2 and 3: Sharpe Ratio versus re-estimation frequency

- *Class boundary setting 2 (the 'sell' and 'buy' class are twice as large as the 'hold' class)*
  2 weeks time horizon: The portfolio performances are negative for the higher split minimums. The Sharpe Ratios are positive for the split minimums smaller than 35, but all below the 0.5 level. For those models we observe again increasing Sharpe Ratios if the re-estimation frequency is higher (with exception of the model with a split minimum of 15).
  4 weeks time horizon: The portfolio performances for these models are all positive. If we again exclude the model with a split minimum of 50 we observe for the models with semi-annually re-estimation frequency and quarterly re-estimation frequency only Sharpe Ratios above the 0.5 level. Although these Sharpe Ratios are lower than for the models with class boundary setting 1, we can conclude that these high Sharpe Ratios are robust to changes in the class boundaries.
- *Class boundary setting 3 (the 'mid' class is twice as large as the 'sell' and 'buy' class)*
  2 weeks time horizon: Portfolio performances have worsened dramatically. 4 of the 18 Sharpe Ratios are negative and none is above the 0.5 level.
  4 weeks time horizon: Portfolio performances are dramatic. Just 6 of the 12 Sharpe Ratios are positive (the highest is 0.12). We can conclude from this test that the FACT algorithm does not succeed in forecasting only extreme interest rate movements. It needs more information on smaller changes in the interest rate to determine whether we might expected changes of the interest rate within a month time.

To conclude this paragraph we argue that the FACT with Gini index model is robust (enough) for changes in the class borders as long as the group of 'hold' events is not too dominant. If the borders are changed to drastically in that way the models will fail to give positive performances. A very nice result of this sensitivity analysis with respect to class borders is that we found very

good performance improvements for the both time horizons. Also very important is that we have found that portfolio performance increases for higher re-estimation frequencies.
We think that based on all sensitivity analyses in this section that the FACT model with 3 classes is a good model to get alpha-performance whether ones to take bets for a horizon of 2 weeks or 4 weeks. To test whether this conclusion is correct we will look at the portfolio performance of these model for the period July 2007-January 2008 which is not included in our research so far.

Section 2. Out-of-sample portfolio performances

In our analysis we have reduced the number of models that are useful in predicting interest rate movements for investment purposes to just two. These two are:

- FACT algorithm with 3 classes and 2 weeks time horizon
- FACT algorithm with 3 classes and 4 weeks time horizon

The selection of the two is based on statistical measures (Chapter 3) and portfolio performances (Chapter 4) that are all based on a data set containing data of the period January 1999-June 2007. To test whether our analysis has resulted in the selection of two useful models we will look at the portfolio performance of the two models over the period July 2007-January 2008. During this period large chocks in financial markets are found as result of the problems with subprime mortgages (this period is called the credit crunch), and the (upcoming) recession in the United States.

In the previous section we found that the higher the re-estimation frequency the better the portfolio performance of the model. We argued that this is the result of 'keeping the ends of the tree up-to-date'. If this statement holds it will also be tested for the period July 2007-January 2008. In the previous section we also did find that the highest Sharpe Ratios were found for models with a split minimum of 10, which is our smallest choice for the split minimum. We will check if we find the same result for the new data set. We did find such good Sharpe Ratios for the FACT algorithm when we used an equal distribution over the classes. Therefore we have taken the class borders such that we have an equal distribution over the classes for this out-of-sample test.

In Appendix E the positions taken every week together with the corresponding realized return can be found for all the different models. In table 9 we give an overview of the results with the absolute return realized over the seven months and the corresponding Sharpe Ratio.

3 classes 2 weeks	quarterly re-est.		semi-annualy re-est.		no re-est.	
	return	Sh. Ratio	return	Sh. Ratio	return	Sh. Ratio
splitmin 10	4,35%	0,58	-9,28%	-1,25	-8,08%	-1,06
splitmin 15	-4,72%	-0,58	-9,28%	-1,25	-8,08%	-1,06
splitmin 20	-3,76%	-0,46	-4,35%	-0,57	-6,60%	-0,85
splitmin 25	0,09%	0,01	-3,46%	-0,42	-5,47%	-0,65
splitmin 35	2,10%	0,25	-3,46%	-0,42	-5,47%	-0,65
splitmin 50	-1,18%	-0,14	-14,15%	-1,73	-19,43%	-2,44

3 classes 4 weeks	quarterly re-est.		semi-annualy re-est.		no re-est.	
	return	Sh. Ratio	return	Sh. Ratio	return	Sh. Ratio
splitmin 10	11,80%	1,83	6,12%	0,93	-0,79%	-0,12
splitmin 15	11,80%	1,83	6,12%	0,93	-0,79%	-0,12
splitmin 20	7,53%	1,11	-7,60%	-0,97	-14,51%	-1,93
splitmin 25	7,53%	1,11	-7,60%	-0,97	-14,51%	-1,93
splitmin 35	7,53%	1,11	-7,60%	-0,97	-14,51%	-1,93
splitmin 50	-0,33%	-0,05	-7,60%	-0,97	-14,51%	-1,93

Table 9. Out-of-sample results

We will discuss the results per model:
3 classes and 2 weeks time horizon: Only positive returns are earned if the trees are re-estimated quarterly. The returns over the different split minimums differ significantly. Remarkable is that we find the two highest Sharpe Ratios, split minimum of 10 and 35, were also in our sensitivity analysis found for these two split minimums. When we look at the returns per quarter (see appendix) then we find negative returns for all split minimums in the third quarter of 2007 and the same (positive) return for the first month of 2008 for all split minimums. So, the differences in returns are mainly the result of the difference in return in the fourth quarter of 2007. Now we also have the position taken as result of the interest rate forecast per split minimum, we can clearly see that the diversity in positions decrease when the split minimum increases. This can be explained by the fact that if a tree is relative small (i.e. large split minimum) there are fewer paths to different forecasts and thus one will often observe the same forecast if no extreme changes in economic data are found.
3 classes and 4 weeks time horizon: Again returns increase when the trees are more frequently re-estimated. The smaller the split minimum the higher the return (and Sharpe Ratio), but it does not change per split minimum change, which is very good with respect to robustness. Apparently, for the model with 4 weeks time horizon the portfolio performance is not heavily dependent on the split minimum chosen. We also observe almost no changes in the forecasts as the split minimum increases. Furthermore we see that the performance of the model with 4 weeks time horizon for these seven months is better than for the model with 2 weeks time horizon, just as we had for the original data set in § 4.1.4.

This out-of-sample test has confirmed again that the FACT models with 3 classes are a good choice for Bund investment purposes, *but only if the models are re-estimated at a quarterly re-estimation frequency*. Recent developments in financial markets are thus very incorporate as learning in FACT algorithm. We also observed that forecasts change less often as the split minimum in the model increases. From an investment perspective it is not desired that a model gives the same signal over a long period of time. Therefore we argue that it is better to have a small split minimum in the FACT algorithm. Based on the performance over the complete period January 1999-January 2008 we suggest a split minimum of 10. If we have to choose between the FACT model with a time horizon of 2 weeks of with 4 weeks, we would choose for a time horizon of 4 weeks. Apparently, the FACT model is most successful in forecasting interest rate movement over the longest time horizon. Thus, to conclude we suggest the FACT algorithm with 3 classes, 4 weeks time horizon, split minimum of 10, and class borders chosen such that class have an equal number of observations as the most useful model for 10-Year Bund investment purposes.

Chapter 5. Conclusion

Goal of this Master Thesis was to investigate whether classification trees are useful for 10 Year Bund investment purposes. We started with a description of what classification trees actually are and how they are constructed. Classification trees assign a class outcome to the values of a set of variables. In our research economic data and risk indicators are used to classify 10-Year Bund interest rate movements. We discussed the CART, QUEST, FACT, and CRUISE algorithms to construct classification trees. These algorithms can be categorized as binomial trees and multiway trees. In binomial trees, the CART and QUEST algorithms, a node in the tree is split in two nodes. In multiway trees, the FACT and CRUISE algorithms, each node is divided into a number of nodes equal to the number of classes possible.

The next step was to determine which classification algorithm was the most successful in forecasting interest rate movements. Unfortunately, for each algorithm a large number of different degrees of freedom are possible within classification models. We considered the following degrees of freedom: 1.) what type of data set to use; 2.) the time horizon over which to forecast the interest rate movements; 3.) the number of classes to divide interest rate movements into; and 4.) the split minimum to use in the classification trees. To reduce the number of possible model settings effectively we decided to first look at different statistical measures like the error rate. At an early stage we could drop the QUEST algorithm, because it clearly underperformed compared to CART, the other binomial tree algorithm. Also the number of degrees of freedom in the models was easily reduced. Most important result is that we chose to use a dataset with weekly relative differences of economic data and both relative differences and absolute levels of certain risk indicators like volatility levels. A second result we found is that classification models are more successful forecasting interest rate movements over longer time horizons (2 and 4 weeks) than shorter time horizons (1 week).

The last stage of our research focused on the robustness of the selected classification models with respect to portfolio performances. We used Sharpe Ratios to measure the portfolio performance for investments in the 10-Year Bund that are based on the interest rate movement forecasts in the models. First we looked at the Sharpe Ratios in the standard models. Secondly, we introduced different non-symmetric misclassification matrices. The use of non-symmetric classification costs did not improve Sharpe Ratios. At this stage of the research we were able to make a decision about which classification tree algorithm had the best portfolio performance. We dropped the CART and CRUISE algorithms and continued our research with the FACT algorithm only. For the FACT algorithm we found that our initial choice of using a expanding window for the learning sample was a good one compared to the use of a rolling window. Next, to test whether portfolio performances of the FACT models were robust to changes in the borders used to classify interest rate movements, we suggested some different class borders. A second objective was to discover if portfolio performance could be improved by using other class borders. We got superb results for the class borders chosen such that all classes have an equal number of observations in the learning sample. Sharpe Ratios far above the desired 0.5 level were found. Best performing were the FACT models with 3 classes and time horizons of 2 and 4 weeks. As a final test we looked at the performance of these models over the period July 2007-January 2008 that was not included in our research so far. If the models were re-estimated quarterly and the split minimum was low we still found good portfolio performances over this instable period in financial markets.

Finally, we believe we can say that there is indeed a classification algorithm that might be useful for 10-Year Bund investment purposes. We conclude that the FACT algorithm with 3 classes, 4 weeks time horizon, split minimum of 10, and class borders chosen such that class have an equal number of observations is the most useful classification model.

Of course there are some drawbacks to our method of research. Most importantly is that we have no guarantee that our findings are the result of overfitting. We have tested four classification algorithms with so many degrees of freedoms in the model settings that it is clearly very likely that we were going to find one model that would give good portfolio performances. On the other hand we did find good portfolio performances over the period July 2007-January 2008 for the models selected as the most useful over the period January 1999-June 2007. For those selected models we were also able to find reasonable explanations why a tree should be re-estimated frequently and why the split minimum should not be large. A second drawback of our method of research is that we have found a classification model that is only suited for forecasting 10-Year Bund interest rate movements. There is no general framework to determine what a good model choice is for interest rates with other maturities. Finally, the classification of the interest rate movements is arbitrarily and up to the user to choose together with the corresponding class borders. Though, a class border choice such that the number of classes assigned to each class in the learning sample is equal, seems reasonable and turned out to be a very successful one in our research.

Further research should be done on the classification of movements of interest rates with other maturities. If the FACT algorithm would also be useful for those classifications, it would strengthen our conclusion that classification algorithms can be helpful to determine Bund investments. For the FACT algorithm itself, more research is needed to find a rule (of thumb) to determine an optimal split minimum. A closer look at the stability of classification trees when they are re-estimated is also necessary. The more robust they are, the more confidence one will have in classification tree forecasts. At last, if a longer data series come available, the effectiveness of the use of an expanding window for the learning sample could be questioned.

References

Breiman, L., Friedman, J., Olshen, R., and Stone, C., *Classification and Regression Trees*, Chapman & Hall, 1984, New York, NY

Lim, Loh, and Shih, *A Comparison of Prediction Accuracy, Complexity, and Training Time of Thirty-Three Old and New Classification Algorithms* Machine Learning, 2000, volume 40, pp. 203-229

Loh, W.-Y., and Shih, Y.-S., *Split Selection Methods for Classification Trees*, Statistica Sinica, 1992, volume 7, pp. 815-840

Loh, W.-Y., *Regression Trees With Unbiased Variable Selection and Interaction Detection*, Statistica Sinica, 2002, Volume 12, pp. 361-386

Kass, G.V., *An Explanatory Technique for Investigating Large Quantities of Categorical Data*, Applied Statistics, 1980, volume 29, pp. 119-127

Loh, W.-Y., and Vanichsetakul, N., *Tree-structured Classification via Generalized Discriminant Analysis*, Journal of the American Statistical Association, 1988, volume 83, pp. 715-725

Kim, H., and Loh, W.-Y., *Classification Trees With Unbiased Multiway Splits*, Journal of the American Statistical Association, 201, volume 96, pp. 598-604

Appendix A. Statistical tests and definitions

ANOVA F-statistic:
A ratio of the Between Group Variation divided by the Within Group Variation

$$F = \frac{Between}{Within} = \frac{MSG}{MSE} = \frac{\sum n_i \left(\bar{x}_i - \bar{x}\right)^2}{\sum (n_i - 1)\left(x_{ij} - \bar{x}_i\right)^2} \frac{I-1}{N-I}$$

with:
n = number of individuals all together
I = number of groups
$\bar{X}$ = mean of entire data set
$\bar{X}$ *Group i has:*
n_i = # of individuals in group i
x_{ij} = value for individal j in group i
$\bar{x}_i$ = mean for group I
$\bar{x}_i$ s_i = standard deviationn for group i

Contingency table:
In statistics contingency tables are used to record and analyse the relationship between two or more variable, most usually categorical variables.

Pearson's chi-square test:
Tests a null hypothesis that the relative frequencies of occurrence of observed events follow a specified frequency distribution. The events are assumed to be independent and have the same distribution, and the outcomes of each event must be mutually exclusive.
Chi-square is calculated by finding the difference between each observed and theoretical frequency for each possible outcome, squaring them, dividing each by the theoretical frequency, and taking the sum of the results. The number of degrees of freedom is equal to the number of possible outcomes, minus 1:

$$X^2_{n-1} = \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{E_i},$$

where
O_i = an observed frequency
E_i = an expected (theoretical) frequency, asserted by the null hypothesis
n = the number of possible outcomes of each event

A test of independence assesses whether paired observations on two variables, expressed in a contingency table, are independent of each other.
A chi-sqaure probability of 0.05 or less is commpnly interpreted as justification for rejecting the null hypothesis that the row variable is unrelated (that is, only randomly related) to the column variable. The alternate hypothesis is accepted that both the variables have an associated relationship.

P-value for F-statistic:
Compare to F(I-1, n-1)-distribution with
I-1 = degrees of freedom in numerator (# groups –1)
n –I = degrees of freedom in denominator (rest of df)
In general one rejects a null hypothesis if the calculated p-value is smaller than the 100%(1-α) level.

P-value for chi-square
Compare to $X((J_t - 1)(M_t - 1))$, $(J_t - 1)(M_t - 1)$ is the degrees of freedom

- J_t the number of classes at node t
- M_t the number of distinct categories in node t (of the categorical variable)

Bonferroni method for multiple comparisons
The more test on a data set are performed, the more likely it is that one rejects the null hypothesis while in fact the hypothesis is true (i.e. a Type I error). This is a consequence of the used methodology: If the number of tests increases, it is more likely to find in one or more cases a rare event such that we reject the null hypothesis. This problem is called the *inflation* of the alpha level. To correct for this one can use the Bonferroni method.

Define: α the significance level = 'probability of making a Type I error'
$1-\alpha$ 'probability of not making a Type I error'
$(1-\alpha)^C$ 'probability of not making a Type I error for a family of C tests'
α[PT] alpha per test = 'probability of making a Type I error when dealing only with one specific test'
α[PF] alpha per family of tests = 'probability of making at least one Type I error for the whole family of tests'
α[PT] = number of significant tests/total number of tests
α[PF] = number of families with at least 1 Type I error/total number of families
$\alpha[PF] = 1 - (1 - \alpha[PT])^C$
$\alpha[PT] = 1 - (1 - \alpha[PF])^{1/C}$

Bonferroni correction of α[PT]) is done by setting the critical signifance level equal to
α[PT]) = α[PC]/C

Levene's F-test:
Levene's F-test is used to test if k samples have equal variances. Equal variances across samples is called homogeneity of variance.
The Levene test is defined as:
Given a variable Y with sample of size N divided into k subgroups, where N_i is the sample size of the ith subgroup, the Levene test statistic is defined as:

$$W = \frac{(N-k)\sum_{i=1}^{k} N_i \left(Z_{i\bullet} - Z_{\bullet\bullet}\right)^2}{(k-1)\sum_{i=1}^{k}\sum_{j=1}^{N_i}\left(Z_{ij} - Z_{i\bullet}\right)^2}$$, where

$Z_{ij} = |Y_{ij} - \overline{Y_i}|$, the mean of the ith subgroup
$Z_{i\bullet}$ the group means of the Z_{ij}
. $Z_{\bullet\bullet}$ the overall mean of the Z_{ij}

For Levene's F-statistic
0% point = 0
50% point = 0.9339
75% point = 1.2964
90% point = 1.7021
95% point = 1.9856
99% point = 2.6109
99.9% point = 3.4789

Appendix B. Classification algorithms

Algorithm 1. Variable selection in QUEST

Let a € (0,1) be a pre-specified level of significance. Assume that X_1, …, X_{K_1} are ordered variables and X_{K_1+1}, …, X_K are categorical variables. Given node t, let $x_{ij}^{(k)}$ denote the value of the k^{th} variable for the i^{th} case in the j^{th} class (i = 1,…, $N_j^{(t)}$; j = 1, …, J_t; k = 1, …, K_1).

1.) If $K_1 \geq 1$, compute the ANOVA F-statistic F_k for each X_k, k = 1, …, K_1. Let k_1 be the smallest integer such that F_{k_1} = max{F_k : k = 1, …, K_1} and define $\hat{\alpha}_1 = \Pr\left\{F_{J_t-1,N(t)-J_t} > F_{k_1}\right\}$, where F_{v_1,v_2} denotes the F-distribution with v_1 and v_2 degrees of freedom.
2.) If $K > K_1$, compute the P-value of the ontingency table chi-square test of independence between class labels and category values for $k = K_1 + 1$, … ,K. The degrees of freedom in each case are given by (n_r –1)x(n_c –1), where n_r and n_c are the numbers of rows and columns of the table with nonzero totals. Let k_2 be the smallest integer such that $\hat{\beta}(k_2) = \min\left\{\hat{\beta}(k) : k = K_1 + 1,..., K\right\}$ and define $\hat{\alpha}_2 = \hat{\beta}(k_2)$.
3.) Define $k' = k_1$ if $\hat{\alpha}_1 \leq \hat{\alpha}_2$; otherwise define k' = k_2.
4.) If $\min(\hat{\alpha}_1, \hat{\alpha}_2) < \alpha / K$, select variable $X_{k'}$ to split the node.
5.) Otherwise. If $\min(\hat{\alpha}_1, \hat{\alpha}_2) \geq \alpha / K$, then
    a. Compute the ANOVA F-statistic $F_k^{(z)}$ (k = 1, …, K_1) for the ordered variables based on the absolute deviations $z_{ik}^{(j)} = | x_{ik}^{(j)} - \bar{x}_k^{(j)} |$, where $\bar{x}_k^{(j)} = N_j(t)^{-1} \sum_{i=1}^{N_j(t)} x_{ik}^{(j)}$. Let k'' be the smallest integer such that $F_{k''}^{(z)}$ = max{$F_k^{(z)}$: k = 1, … ,K_1}.
    b. Compute $\tilde{\alpha} = \Pr\left\{F_{J_t-1,N(t)-J_t} > F_{k''}^{(z)}\right\}$. If $\tilde{\alpha} < \alpha/(K + K_1)$, select variable $X_{k''}$ to split the node. Otherwise select variable $X_{k'}$.

Algorithm 2. Split selection for an ordered variable in QUEST

Let X be the selected variable to split node t.

1.) Apply the 2-means clustering algorithm of Hartigan and Wong (1979) to divide the J_t classes into two superclasses A and B, using the two most extreme sample means as initial cluster centers. If the sample means are identical, let A contain the most populous class and B contain the other classes.
2.) Let $\bar{x}_A$ and s^2_A denote the sample mean and variance of superclass A. Similarly, let $\bar{x}_B$ and s^2_B denote the corresponding quantities for superclass B. Let $p(A | t) = \sum_{j \in A} p(j | t)$ and $p(B | t) = 1 - p(A | t)$ denote the superclass priors.
3.) Take logs on both sides of the equation $p(A | t) s_A^{-1} \phi\left\{\left(x - \bar{x}_A\right) / s_A\right\} = p(B | t) s_B^{-1} \phi\left\{\left(x - \bar{x}_B\right) / s_B\right\}$ to obtain the quadratic equation $ax^2 + bx + c = 0$, where
    a. $= s^2_A - s^2_B$

b. $= 2\left(\bar{x}_A \cdot s_B^2 - \bar{x}_B \cdot s_A^2\right)$

c. $= \left(\bar{x}_B \cdot s_A\right)^2 - \left(\bar{x}_A \cdot s_B\right)^2 + 2 s_A^2 s_B^2 \log\left[\{p(A \mid t) s_B\} / \{p(B \mid t) s_A\}\right]$.

If $a = 0$ and $\bar{x}_A \neq \bar{x}_B$, there is only one root given by $x = \left(\bar{x}_A + \bar{x}_B\right)/2 - \left(\bar{x}_A - \bar{x}_B\right)^{-1} s_A^2 \log\{p(A \mid t)/p(B \mid t)\}$.

The equation has no roots if $a = 0$ and $\bar{x}_A = \bar{x}_B$.

4.) The node is split at $X = d$ where d is defined as follows:

a. If $a = 0$ then $d\begin{cases}\left(\bar{x}_A + \bar{x}_B\right)/2 - \left(\bar{x}_A - \bar{x}_B\right)^{-1} s_A^2 \log\{p(A \mid t)/p(B \mid t)\} & , \bar{x}_A \neq \bar{x}_B \\ \bar{x}_A & , \bar{x}_A = \bar{x}_B.\end{cases}$

b. Else, if $a \neq 0$, then:

i. If $b^2 - 4ac < 0$, define $d = \left(\bar{x}_A + \bar{x}_B\right)/2$. It can be verified that $b^2 - 4ac \geq 0$ if P(A| t) = P(B| t).

ii. Else, If $b^2 - 4ac \geq 0$, then:

1. Define d to be the root $(2a)^{-1}\{-b \pm \sqrt{b^2 - 4ac}\}$ that is closer to $\bar{x}_A$, provided this yield two nonempty nodes.

2. Otherwise, define $d = \left(\bar{x}_A + \bar{x}_B\right)/2$.

<u>Algorithm 3. Split selection for a categorical variable in QUEST</u>

Suppose X is a categorical variable taking values in the set $\{c_1, \ldots, c_M\}$.

1.) Transform each value of X into an M-dimensional dummy column vector **v** = $(v_1, \ldots, v_M)'$, where $v_1 = \begin{cases} 1, & , X = c_l \\ 0, & otherwise \end{cases}$

Let **V** be the N x M data matrix consisting of the v-values.

2.) Let **I** denote the N x N identify matrix and let **1** be an N-column of 1's. Let **H** = **I** – N^{-1}**11'** denote the centering matrix and obtain the singular value decomposition **HV** = **PDQ'** with **D** = diag($d_1, \ldots, d_M$) such that $d_1 \geq \ldots \geq d_M \geq 0$.

3.) Let ε be the machine precision, i.e., the smallest floating point number such that if $u = 1 + ε$, then $u > 1$. Define an eigenvalue d_m as 'positive' if it satisfies $d_m > \max(M, N) d_1 ε$, and as 'zero' otherwise (Mathworks, 1991). The rank r of **T** is defined to be the number of 'positive' eigenvalues. Let **F** denote the M x r submatrix of **Q** consisting of its first r columns and let **U** = diag($d_1^{-1}, \ldots, d_r^{-1}$).

4.) Reduce the dimension of **v** by the transforming it to **y** = **F'v**.

5.) Define for each j the M x N_j matrix $\mathbf{L}_j = \left(\bar{v}^{(j)} - \bar{v}, \ldots, \bar{v}^{(j)} - \bar{v}\right)$ and let **G** bet the N x M matrix **G** = $(\mathbf{L}_1, \ldots, \mathbf{L}_J)'$, so that **B** = **G'G**. Perform a singular value decomposition of the matrix **GFU** and let **a** be the eigenvector associated with the largest eigenvalue.

6.) Transform each **v** to ζ = **a'UF'v**. This maps each c_l to a ζ-value.

7.) Apply algorithm 1 to the ζ data values to split the node.

8.) Re-express a split of the form '$ζ \leq ζ_0$' to the form '$X \in A$'.

Algorithm 4. CRIMCOORD transformation for categorical variables

1.) Suppose X takes categorical values $a_1, ..., a_c$.
2.) Define V = (v_1, ..., v_c) where $v_i = I(X = a_i)$.
3.) Transform the V-vectors to principal component axes and drop components with near zero variance.
4.) Project the reduced-dimensional V-data onto the largest contingency crimcoord ζ.

Replace X with the real-valued ζ in the rest of the algorithm.

Algorithm 5. Splitting algorithm 1D in CRUISE

Let a be a selected significance level (default is 0.05). Suppose $X_1, ..., X_{K_1}$ are numerical and $X_{K_1+1}, ..., X_K$ are categorical variables.

1.) Carry out an ANOVA analysis on each numerical variable and compute its p-value. Suppose X_{K_1} has the smallest p-value $\hat{\alpha}_1$.
2.) For each categorical variable, form a contingency table with the categorical values as rows and class values as columns and find its χ^2 p-value. Let the smallest p-value be $\hat{\alpha}_2$ and the associated variable be X_{K_2}.
3.) Define $k' \begin{cases} k_1, & \hat{\alpha}_1 \le \hat{\alpha}_2 \\ k_2, & \hat{\alpha}_1 > \hat{\alpha}_2. \end{cases}$
4.) If $\min(\hat{\alpha}_1, \hat{\alpha}_2) < \alpha / K$ (first Bonferroni correction), choose $X_{k'}$ as the split variable.
5.) Otherwise, find the p-value for Levene's F-test on absolute deviations about the class mean foe each numerical variable. Suppose $X_{k''}$ has smallest p-value $\tilde{\alpha}$.
    a. If $\tilde{\alpha} < a/(K + K_1)$, choose $X_{k''}$ (second Bonferroni correction)
    b. Otherwise, choose $X_{k'}$.

Algorithm 6. Splitting algorithm 2D in CRUISE

Suppose $X_1, ..., X_{K_1}$ are numerical and $X_{K_1+1}, ..., X_K$ are categorical variables. Let J_t be the number of classes represented at node t.

1.) Marginal test for each numerical variable X:
    a. Divide the data into four groups at the sample quartiles of X.
    b. Construct a J_t x 4 contingency table with classes as rows and groups as columns.
    c. Compute the Pearson χ^2 statistic with $\nu = 3(J_t - 1)$ degrees of freedom.
    d. Convert χ^2 to an appropriate standard normal value with the Peizer-Pratt transformation
    $$z = \begin{cases} |W|^{-1}\,(W - 1/3)\sqrt{(\nu - 1)\log\{(\nu - 1)/\chi^2\} + W} & ,\nu > 1 \\ \sqrt{\chi^2} & ,\nu = 1 \end{cases} \qquad (1)$$
    where $W = \chi^2 - \nu + 1$.
    Let z_n denote the largest among the K_1 z-values.
2.) Marginal test for each categorical variable X: Let C denote the number of categories of X.
    a. Construct a J_t x C contingency table with classes as rows and the C categories as columns,
    b. Compute the Pearson χ^2 statistic with $(J_t - 1)(C - 1)$ degrees of freedom.
    c. Use the Peizer-Pratt transformation (1) to convert it to a z-value.
    Let z_c denote the largest among the $(K - K_1)$ z-values.
3.) Interaction test for each pair of numerical variables (X_k, $X_{k'}$):
    a. Divide the (X_k, $X_{k'}$) space into four quadrants as the sample medians.
    b. Construct a J_t x 4 contingency table with classes as rows and the quadrants as columns.
    c. Compute the Pearson χ^2 statistic with $3(J_t - 1)$ degrees of freedom.

d. Use the Peizer-Pratt transformation (1) to convert t to a z-value. Let z_{nn} denote the largest among the $K_1(K_1 - 1)/2$ z-values.

4.) Interaction test for each pair of categorical variables: Use pairs of categorical values to form the groups in the table. If the pair of variables take C_1 and C_2 categorical values, a J_t x C_1C_2 table is obtained. Let z_{cc} denote the largest among the $(K - K_1)(K - K1 - 1)/2$ z-values.

5.) Interaction tests for pairs $(X_k, X_{k'})$ where X_k is numerical and $X_{k'}$ is categorical: If $X_{k'}$ takes C values, obtain a J_t x $2C$ table. Let z_{nc} denote the largest among the $K_1(K - K_1)$ z-values.

Let f^* be the bootstrap value from Algorithm 7 (Bootstrap bias correction) and define $z^* = \max\{f^* z_n, z_c, f^* z_{nn}, z_{cc}, z_{nc}\}$.

1. If $f^*z_n = z^*$, select the numerical variable with the largest z.
2. If $z_c = z^*$, select a categorical variable with the largest z.
3. If $f^*z_{nn} = z^*$, select the numerical variable in the pair with the larger z.
4. If $z_{cc} = z^*$, select the categorical variable in the pair with the larger z.
5. If $z_{nc} = z^*$, select the categorical variable in the interacting pair.

Algorithm 7. Bootstrap bias correction

1.) Create a bootstrap learning sample by copying the values of the variable and bootstrapping the Y columns so that the response variable is independent of the predictors.
2.) Apply steps 1-5 in Algorithm 2 (Splitting algorithm 2D) to the bootstrap sample to get five sets of z-values.
3.) Given $f > 1$, select a numerical variable if f $\max\{z_n, z_{nn}\} \geq \max\{z_c, z_{cc}, z_{nc}\}$. Otherwise, select a categorical variable.
4.) Repeat steps 1-3 many times with several values of f. Let $\pi(f)$ be the proportion of times that a numerical variable is selected.
5.) Linearly interpolate if necessary to find f^* such that $\pi(f^*)$ equals the proportion of numerical variables in the data.

Algorithm 8. Box-Cox transformation

Suppose X is the selected variable. If X is categorical, its values are first transformed to crimcoord values.

1.) Let $x_{(i)}$ denote the i^{th} order statistic. Define $\theta = 0$ if $x_{(1)} > 0$ and $\theta = 2x_{(1)} - x_{(2)}$ otherwise.
2.) Given λ, define $x^{(\lambda)} = \begin{cases} \left[(x-\theta)^{\lambda} - 1\right]/\lambda & ,if \;\; \lambda \neq 0 \\ \log(x-\theta) & if \;\; \lambda = 0. \end{cases}$
3.) Let $\hat{\lambda}$ be the minimizer of $\sum_j \sum_i \left[x_{ji}^{(\lambda)} - \overline{x}_j^{(\lambda)}\right]^2 \exp\left\{-2n^{-1}\lambda\left[\sum_j \sum_i \log(x_{ji})\right]\right\}$ where x_{ji} is the i^{th} value of X in class j and $\overline{x}_j^{(\lambda)}$ is the sample class mean of their transformed values.
4.) Transform each x value to $x^{(\hat{\lambda})}$.

Appendix C. Selected Variables

M = Monthly available data
W =Weekly available data
RI = Risk indicator (available at weekly frequency)

GER = Germany
EUR = Euro zone
US = United States
Em. Mar = Emerging markets

Change = absolute change taken as variable
Stationair (xx) = Equilibrium level is xx, absolute change taken as variable
Absolute = absolute value taken as variable
Trend = variable contains trend, relative difference taken as variable
MoM = monthly change taken as variable
YoY = yearly change taken as variable

Name	Start data	Month/Week/RI	GER/EUR/US	Quote type	Description
EUBCI	jan-95	M	EUR	Change	Economy Indicator
EUESEMU	jan-95	M	EUR	Stationair (100)	Economic Sentiment
EUCCEMU	jan-95	M	EUR	Absolute	Consumer Confidence
CPEXEMU	mrt-96	M	EUR	Trend	Consumer Prices
PPTXEMU	jan-95	M	EUR	Trend	Producer Prices Industry
EUITEMU	jan-95	M	EUR	Trend	Industrial Production
CPALEMU	mrt-96	M	EUR	Trend	Harmonized Consumer Prices
ECMSM3	jan-95	M	EUR	Trend	Money Supply M3
OLEDEU12	jan-95	M	EUR	Trend	OECD Leading Indicators EU12
GRIMP95	jan-95	M	GER	Stationair (100)	Import Prices
GRIFPBUS	jan-95	M	GER	Stationair (100)	Business Climate
GRIFPEX	jan-95	M	GER	Stationair (100)	Business Expectations
PCE CONC	jan-95	M	US	Trend	Personal Consumption Expenditures
PITL	jan-95	M	US	Trend	Personal Income (SAAR)
PCE CORE	jan-95	M	US	Trend	Personal Consumption Expenditures
NFP TCH	jan-95	M	US	Absolute	Employees on Nonfarm payrolls
IP CHNG	jan-95	M	US	MoM	Industrial Production
DGNOTOT	jan-95	M	US	Trend	Durable Goods New Orders
NAPMNEWO	jan-95	M	US	Stationair (50)	Manufacturing
NAPM PMI	jan-95	M	US	Stationair (50)	Manufacturing
NAPMBACK	jan-95	M	US	Stationair (50)	Manufacturing
USHBMIDX	jan-95	M	US	Stationair (50)	Home Builders Index
GRIFOBSI	jan-95	M	EUR	Stationair (100)	Business Climate
GEINYY	jan-95	M	GER	YoY	Industrial Production
RSSAEMUM	apr-95	M	EUR	MoM	Retail Sales Volume
RSWAEMUY	apr-96	M	EUR	YoY	Retail Sales Volume
GRCP20MM	jan-95	M	GER	MoM	Consumer Prices
GRCP20YY	jan-95	M	GER	YoY	Consumer Prices
GRCP2HMM	apr-95	M	GER	MoM	Harmonized Consumer Prices
GRCP2HYY	mrt-96	M	GER	YoY	Harmonized Consumer Prices
NAPMPRIC	jan-95	M	US	Stationair (50)	Purchasing costss
GRZEWI	jan-95	M	GER	Change	Economic Expectations
GRZECURR	jan-95	M	GER	Change	Economic Sentiment
PPI CHNG	jan-95	M	US	Change	Producer Prices
PPI YOY	jan-95	M	US	YoY	Producer Prices
PXFECHNG	jan-95	M	US	Change	Producer Prices
PPI XYOY	jan-95	M	US	YoY	Producer Prices
RSTAMOM	jan-95	M	US	MoM	Adjusted Retail&Food Services
MTIBCHNG	jan-95	M	US	Change	Manufacturing&Trade
GRFRIAMM	jan-95	M	GER	MoM	Retail Sales Constant
GRFRINYY	jan-95	M	GER	YoY	Retail Sales Constant

EUICEMU	jan-95	M	EUR	Change	Manufacturing
EUSCEMU	may-95	M	EUR	Change	Services
PITLCHNG	jan-95	M	US	Change	Personal Income
PCE CRCH	jan-95	M	US	Change	Personal Consumption Expenditure
PCE DEFY	jan-95	M	US	Change	Personal Consumption Expenditure
PCE CMOM	jan-95	M	US	MoM	Personal Consumption Expenditure
PCE CYOY	jan-95	M	US	YoY	Personal Consumption Expenditure
GRUECHNG	jan-95	M	GER	Change	Unemployment
GRUEPR	jan-95	M	GER	Stationair (10)	Unemployment
TMNOCHNG	jan-95	M	US	Change	Manufacturing New Orders
UMRTEMU	jan-95	M	EUR	Change	Unemployment
USURTOT	jan-95	M	US	Rate	Unemployment
NAPMNMAN	aug-97	M	US	Stationair (50)	Non-Manufacturing
EUITEMUM	jan-95	M	EUR	MoM	Industrial Production
EUIPEMUY	jan-95	M	EUR	YoY	Industrial Production
GRIFPCA	jan-95	M	GER	Stationair (100)	Economic Sentiment
NHSLTOT	jan-95	M	US	Absolute	New (1-familiy) Houses Sold
NHSLCHNG	jan-95	M	US	Change	New (1-familiy) Houses Sold
GRPFIYOY	jan-95	M	GER	YoY	Producer Prices
GRPFIMOM	jan-95	M	GER	MoM	Producer Prices
IMP1CHNG	jan-95	M	US	Change	Import Prices
IMP1YOY%	jan-95	M	US	YoY	Import Prices
CONCCONF	jan-95	M	US	Stationair (100)	Consumer Confidence
SX5E	jan-95	W	US	Trend	Euro STOXX 50
M1	jan-95	W	US	Trend	Money Supply M1
US0003M	jan-95	W	US	Interest Rate	3M Interest Rate
SPX	jan-95	W	US	Trend	S&P 500
GSINER	jan-95	W	US	Trend	Industrial
CRY	jan-95	W	WORLD	Trend	Commodity market
CL1	jan-95	W	WORLD	Trend	Oil
GC1	jan-95	W	WORLD	Trend	Gold
GETB1	jan-95	W	GER	Interest Rate	6 Month rate
EU0003M	jan-95	W	EUR	Interest Rate	3M Interest Rate
INJCJC	jan-95	W	US	Nominal	Initial Job Claims
INJCSP	jan-95	W	US	Nominal	Continuing Job Claims
EURR002W	jan-99	W	EUR	Interest Rate	Refi
ACNFCOMF	jan-95	W	US	Stationair (0)	ABC News\Washington Post US Survey
EURUSD	jan-99	W	EUR	Change	EUR/USD
EURGBP	jan-99	W	EUR	Change	EUR/GBP
EURJPY	jan-99	W	EUR	Change	EUR/JPY
JPMVXYG7	jan-99	RI	G7	Nominal	3-month FX volatility G7
VIX	jan-99	RI	US	Nominal	US equity implied volatility (ie. VIX)
MSELEGF	jan-99	RI	Em. Mar.	Trend	EM Equities
EUSS10	jan-99	RI	EUR	Change	Swap spreads
USSP10	jan-99	RI	US	Change	Swap spreads
JYSS10	jan-99	RI	Japan	Change	Swap spreads
JPEMSOSD	jan-99	RI	Em. Mar.	Change	EM Bond Spreads (over Treasuries)
EURUSDV1Y CMPN	jan-99	RI	Euro/Dollar	Volatility level	1-year FX volatility (GHI)
EURJPYV1Y CMPN	jan-99	RI	Euro/Yen	Volatility level	1-year FX volatility (GHI)
USDJPYV1Y CMPN	jan-99	RI	Dollar/Yen	Volatility level	1-year FX volatility (GHI)

Appendix D. Statistical measures FACT: weekly data and risk indicators

Upup = 'strong sell'-event; Up = 'sell'-event;
Down = 'buy'-event; Downdown = 'strong buy'-event
3c = 3 classes; 4c = 4 classes; 5c = 5 classes; WDRI = Weekly Data and Risk Indicators
1W = 1 week time horizon; 2W = 2 week time horizon; 4W = 4 week time horizon;
Note that all number are the average of 10 sample of 10-fold crossvalidation

ERROR RATE

	min split 50	min split 40	min split 30	min split 25	min split 20	min split 15	min split 10
3c_WDRI_1W	0,6009	0,6007	0,5989	0,5978	0,6108	0,6088	0,6198
3c_WDRI_2W	0,6474	0,6452	0,6549	0,6366	0,6533	0,6527	0,6535
3c_WDRI_4W	0,5869	0,5788	0,5960	0,6035	0,6099	0,6124	0,6104
4c_WDRI_1W	0,7125	0,7147	0,7152	0,7158	0,7180	0,7266	0,7295
4c_WDRI_2W	0,7333	0,7328	0,7311	0,7249	0,7209	0,7502	0,7320
4c_WDRI_4W	0,7286	0,7173	0,7239	0,7217	0,7128	0,7144	0,7199
5c_WDRI_1W	0,7429	0,7475	0,7530	0,7517	0,7528	0,7549	0,7444
5c_WDRI_2W	0,7238	0,7410	0,7355	0,7408	0,7348	0,7414	0,7526
5c_WDRI_4W	0,7509	0,7456	0,7509	0,7584	0,7608	0,7681	0,7812

ERROR RATE UP

	min split 50	min split 40	min split 30	min split 25	min split 20	min split 15	min split 10
3c_WDRI_1W	0,2670	0,2873	0,2808	0,2701	0,2789	0,2783	0,2724
3c_WDRI_2W	0,3405	0,3226	0,3304	0,3327	0,3548	0,3339	0,3312
3c_WDRI_4W	0,2243	0,2174	0,2267	0,2322	0,2424	0,2378	0,2388

CORRECT RATE UP

	min split 50	min split 40	min split 30	min split 25	min split 20	min split 15	min split 10
3c_WDRI_1W	0,2897	0,2566	0,2753	0,2913	0,2760	0,2752	0,2737
3c_WDRI_2W	0,3552	0,3631	0,3544	0,3822	0,3602	0,3619	0,3731
3c_WDRI_4W	0,4093	0,4252	0,4143	0,4118	0,3934	0,3904	0,4043

ERROR RATE DOWN

	min split 50	min split 40	min split 30	min split 25	min split 20	min split 15	min split 10
3c_WDRI_1W	0,2828	0,2643	0,2763	0,2933	0,2760	0,2791	0,2728
3c_WDRI_2W	0,3099	0,3022	0,3205	0,2902	0,3136	0,3115	0,2969
3c_WDRI_4W	0,1980	0,2043	0,2140	0,2234	0,2238	0,2516	0,2409

CORRECT RATE DOWN

	min split 50	min split 40	min split 30	min split 25	min split 20	min split 15	min split 10
3c_WDRI_1W	0,2710	0,3129	0,3024	0,2912	0,2880	0,2918	0,2790
3c_WDRI_2W	0,3898	0,3988	0,3808	0,3963	0,3729	0,3817	0,3815
3c_WDRI_4W	0,4201	0,4228	0,3880	0,3733	0,3736	0,3595	0,3561

PREDICTION: # UP

	min split 50	min split 40	min split 30	min split 25	min split 20	min split 15	min split 10
3c_WDRI_1W	74	80	85	90	91	91	97
3c_WDRI_2W	164	160	169	158	154	152	157
3c_WDRI_4W	141	135	141	131	133	139	130

PREDICTION: # MID

	min split 50	min split 40	min split 30	min split 25	min split 20	min split 15	min split 10
3c_WDRI_1W	312	306	297	291	278	277	275
3c_WDRI_2W	114	121	115	131	133	133	146
3c_WDRI_4W	186	194	184	198	198	198	208

PREDICTION: # DOWN

	min split 50	min split 40	min split 30	min split 25	min split 20	min split 15	min split 10
3c_WDRI_1W	69	69	73	74	86	88	83
3c_WDRI_2W	177	173	170	165	167	169	151
3c_WDRI_4W	125	123	126	123	121	115	114

PREDICTION: # UPUP

	min split 50	min split 40	min split 30	min split 25	min split 20	min split 15	min split 10
4c_WDRI_1W	47	51	57	58	58	66	67
4c_WDRI_2W	92	91	86	92	86	85	85
4c_WDRI_4W	79	86	77	87	92	93	95

PREDICTION: # UP

	min split 50	min split 40	min split 30	min split 25	min split 20	min split 15	min split 10
4c_WDRI_1W	130	134	131	137	138	139	152
4c_WDRI_2W	117	126	124	127	126	133	135
4c_WDRI_4W	122	133	130	130	127	126	126

PREDICTION: # DOWN

	min split 50	min split 40	min split 30	min split 25	min split 20	min split 15	min split 10
4c_WDRI_1W	243	226	216	201	202	199	177
4c_WDRI_2W	181	175	174	171	174	167	161
4c_WDRI_4W	189	167	181	166	165	164	157

PREDICTION: # DOWNDOWN

	min split 50	min split 40	min split 30	min split 25	min split 20	min split 15	min split 10
4c_WDRI_1W	35	44	51	60	57	52	59
4c_WDRI_2W	65	62	71	65	69	69	73
4c_WDRI_4W	63	66	65	70	69	69	76

PREDICTION: # UPUP

	min split 50	min split 40	min split 30	min split 25	min split 20	min split 15	min split 10
5c_WDRI_1W	46	46	41	37	40	43	43
5c_WDRI_2W	14	18	26	28	30	32	34
5c_WDRI_4W	124	122	118	115	108	109	106

PREDICTION: # UP

	min split 50	min split 40	min split 30	min split 25	min split 20	min split 15	min split 10
5c_WDRI_1W	58	55	67	65	63	70	76
5c_WDRI_2W	70	76	98	94	93	102	106
5c_WDRI_4W	37	39	47	49	50	57	66

PREDICTION: # MID

	min split 50	min split 40	min split 30	min split 25	min split 20	min split 15	min split 10
5c_WDRI_1W	215	207	206	211	203	203	198
5c_WDRI_2W	246	227	210	211	207	195	183
5c_WDRI_4W	97	110	98	104	108	113	104

PREDICTION: # DOWN

	min split 50	min split 40	min split 30	min split 25	min split 20	min split 15	min split 10
5c_WDRI_1W	125	133	123	123	127	117	112
5c_WDRI_2W	120	123	109	106	111	106	111
5c_WDRI_4W	105	100	94	100	102	94	99

PREDICTION: # DOWNDOWN

	min split 50	min split 40	min split 30	min split 25	min split 20	min split 15	min split 10
5c_WDRI_1W	11	14	18	19	22	23	26
5c_WDRI_2W	4	10	12	15	14	19	20
5c_WDRI_4W	90	81	95	84	84	80	77

PREDICTED: DOWN, OBSERVED: UPUP

	min split 50	min split 40	min split 30	min split 25	min split 20	min split 15	min split 10
4c_WDRI_1W	0,1947	0,1914	0,1885	0,1929	0,1842	0,1934	0,1880
4c_WDRI_2W	0,2488	0,2438	0,2436	0,2465	0,2384	0,2575	0,2334
4c_WDRI_4W	0,2034	0,2027	0,2083	0,2048	0,2004	0,2095	0,1951
5c_WDRI_1W	0,1546	0,1380	0,1380	0,1590	0,1546	0,1455	0,1251
5c_WDRI_2W	0,1114	0,1207	0,1237	0,1132	0,1051	0,1031	0,1123
5c_WDRI_4W	0,2310	0,2166	0,2067	0,2383	0,2254	0,2183	0,2139

PREDICTED: DOWNDOWN, OBSERVED: UPUP

	min split 50	min split 40	min split 30	min split 25	min split 20	min split 15	min split 10
4c_WDRI_1W	0,2085	0,2053	0,1963	0,2152	0,1930	0,2079	0,2192
4c_WDRI_2W	0,1700	0,1639	0,1820	0,1657	0,1780	0,1797	0,1897
4c_WDRI_4W	0,1557	0,1816	0,1668	0,1681	0,1580	0,1564	0,2037
5c_WDRI_1W	0,0883	0,1749	0,1443	0,1422	0,1863	0,1408	0,1997
5c_WDRI_2W	0,0889	0,1798	0,2060	0,1275	0,1541	0,1726	0,1651
5c_WDRI_4W	0,1694	0,1994	0,2178	0,1789	0,2044	0,2006	0,2116

PREDICTED: DOWN, OBSERVED: UP

	min split 50	min split 40	min split 30	min split 25	min split 20	min split 15	min split 10
4c_WDRI_1W	0,2746	0,2762	0,2818	0,2822	0,2769	0,2783	0,2797
4c_WDRI_2W	0,2683	0,2675	0,2730	0,2671	0,2577	0,2680	0,2778
4c_WDRI_4W	0,2545	0,2369	0,2480	0,2454	0,2439	0,2398	0,2374
5c_WDRI_1W	0,1965	0,2027	0,1996	0,2101	0,1768	0,1827	0,2087
5c_WDRI_2W	0,2065	0,2182	0,2102	0,2062	0,2082	0,2169	0,2032
5c_WDRI_4W	0,1193	0,1156	0,1091	0,1127	0,1256	0,1306	0,1226

PREDICTED: DOWNDOWN, OBSERVED: UP

	min split 50	min split 40	min split 30	min split 25	min split 20	min split 15	min split 10
4c_WDRI_1W	0,2521	0,2766	0,2678	0,2383	0,2833	0,2856	0,2784
4c_WDRI_2W	0,2629	0,3197	0,3061	0,2800	0,2882	0,2622	0,2722
4c_WDRI_4W	0,2628	0,2633	0,2526	0,2539	0,2166	0,2464	0,2336
5c_WDRI_1W	0,1772	0,2124	0,1503	0,1513	0,1281	0,1598	0,1719
5c_WDRI_2W	0,2111	0,1883	0,1428	0,1486	0,1968	0,2166	0,1527
5c_WDRI_4W	0,1454	0,1266	0,1249	0,1458	0,1499	0,1411	0,1469

PREDICTED: UPUP, OBSERVED: DOWN

	min split 50	min split 40	min split 30	min split 25	min split 20	min split 15	min split 10
4c_WDRI_1W	0,3233	0,3248	0,3066	0,3072	0,3115	0,3152	0,3251
4c_WDRI_2W	0,3169	0,3028	0,2954	0,3068	0,2986	0,2947	0,3115
4c_WDRI_4W	0,2381	0,2223	0,2416	0,2487	0,2425	0,2476	0,2481
5c_WDRI_1W	0,1816	0,2076	0,2282	0,2421	0,2485	0,1863	0,2047
5c_WDRI_2W	0,2093	0,2205	0,1967	0,2075	0,2407	0,2217	0,1563
5c_WDRI_4W	0,1926	0,1814	0,2005	0,1807	0,1837	0,1934	0,2025

PREDICTED: UP, OBSERVED: DOWN

	min split 50	min split 40	min split 30	min split 25	min split 20	min split 15	min split 10
4c_WDRI_1W	0,3275	0,3280	0,3339	0,3385	0,3278	0,3468	0,3261
4c_WDRI_2W	0,3273	0,3253	0,3452	0,3102	0,3109	0,3243	0,3071
4c_WDRI_4W	0,3215	0,3232	0,3139	0,3121	0,2988	0,3041	0,3087
5c_WDRI_1W	0,3034	0,3018	0,2792	0,2818	0,2707	0,2841	0,2763
5c_WDRI_2W	0,2518	0,2489	0,2440	0,2624	0,2427	0,2352	0,2764
5c_WDRI_4W	0,1833	0,1718	0,1882	0,2064	0,2170	0,1805	0,1709

PREDICTED: UPUP, OBSERVED: DOWNDOWN

	min split 50	min split 40	min split 30	min split 25	min split 20	min split 15	min split 10
4c_WDRI_1W	0,1824	0,1946	0,1837	0,1759	0,2058	0,1988	0,2144
4c_WDRI_2W	0,1610	0,1969	0,1790	0,1971	0,1802	0,1898	0,1613
4c_WDRI_4W	0,1484	0,1679	0,1655	0,1552	0,1639	0,1655	0,1694
5c_WDRI_1W	0,1430	0,1052	0,1057	0,1258	0,1039	0,0851	0,1315
5c_WDRI_2W	0,0988	0,1120	0,0980	0,0935	0,0973	0,1075	0,0880
5c_WDRI_4W	0,1563	0,1718	0,1698	0,1904	0,1834	0,1914	0,1987

PREDICTED: UP, OBSERVED: DOWNDOWN

	min split 50	min split 40	min split 30	min split 25	min split 20	min split 15	min split 10
4c_WDRI_1W	0,1825	0,1803	0,1871	0,1869	0,1784	0,1794	0,1859
4c_WDRI_2W	0,2132	0,2130	0,2207	0,2013	0,2156	0,2096	0,2303
4c_WDRI_4W	0,2229	0,2144	0,2112	0,2160	0,2026	0,2100	0,2049
5c_WDRI_1W	0,0845	0,0778	0,0789	0,0774	0,0723	0,0762	0,1024
5c_WDRI_2W	0,1082	0,1046	0,1052	0,1204	0,1216	0,0941	0,1067
5c_WDRI_4W	0,1672	0,1773	0,1801	0,1700	0,1709	0,2106	0,1842

PREDICTED: UPUP, OBSERVED: UPUP

	min split 50	min split 40	min split 30	min split 25	min split 20	min split 15	min split 10
4c_WDRI_1W	0,1802	0,1862	0,2201	0,2189	0,2041	0,2032	0,1718
4c_WDRI_2W	0,2609	0,2514	0,2645	0,2501	0,2640	0,2153	0,2819
4c_WDRI_4W	0,3413	0,3433	0,3290	0,3218	0,3266	0,3120	0,3057
5c_WDRI_1W	0,2098	0,2075	0,1749	0,1717	0,2159	0,1966	0,1892
5c_WDRI_2W	0,1559	0,1699	0,1766	0,1807	0,1501	0,1337	0,1972
5c_WDRI_4W	0,2907	0,2906	0,2859	0,2867	0,2851	0,2837	0,2670

PREDICTED: UP, OBSERVED: UPUP

	min split 50	min split 40	min split 30	min split 25	min split 20	min split 15	min split 10
4c_WDRI_1W	0,2081	0,2157	0,2038	0,1878	0,2154	0,1973	0,2127
4c_WDRI_2W	0,2158	0,2287	0,2208	0,2377	0,2235	0,2424	0,2182
4c_WDRI_4W	0,2513	0,2287	0,2483	0,2514	0,2533	0,2543	0,2483
5c_WDRI_1W	0,1234	0,1396	0,1419	0,1271	0,1364	0,1630	0,1407
5c_WDRI_2W	0,1472	0,1395	0,1518	0,1377	0,1383	0,1510	0,1186
5c_WDRI_4W	0,3028	0,2938	0,2760	0,3228	0,2790	0,2783	0,2929

PREDICTED: UPUP, OBSERVED: UP

	min split 50	min split 40	min split 30	min split 25	min split 20	min split 15	min split 10
4c_WDRI_1W	0,3141	0,2944	0,2896	0,2981	0,2786	0,2829	0,2887
4c_WDRI_2W	0,2612	0,2490	0,2611	0,2460	0,2572	0,3002	0,2454
4c_WDRI_4W	0,2722	0,2665	0,2639	0,2743	0,2670	0,2750	0,2768
5c_WDRI_1W	0,2036	0,2085	0,2140	0,2074	0,2019	0,2420	0,1891
5c_WDRI_2W	0,2241	0,1919	0,2328	0,2183	0,2146	0,2393	0,2916
5c_WDRI_4W	0,1846	0,1942	0,1825	0,1694	0,1714	0,1737	0,1942

PREDICTED: UP, OBSERVED: UP

	min split 50	min split 40	min split 30	min split 25	min split 20	min split 15	min split 10
4c_WDRI_1W	0,2819	0,2760	0,2753	0,2868	0,2784	0,2767	0,2753
4c_WDRI_2W	0,2438	0,2330	0,2133	0,2507	0,2500	0,2237	0,2445
4c_WDRI_4W	0,2043	0,2337	0,2266	0,2205	0,2454	0,2317	0,2381
5c_WDRI_1W	0,2120	0,2029	0,2208	0,2265	0,2398	0,2092	0,2070
5c_WDRI_2W	0,2420	0,2415	0,2177	0,2200	0,2205	0,2206	0,2085
5c_WDRI_4W	0,1536	0,1599	0,1642	0,1389	0,1399	0,1580	0,1431

PREDICTED: DOWN, OBSERVED: DOWN

	min split 50	min split 40	min split 30	min split 25	min split 20	min split 15	min split 10
4c_WDRI_1W	0,3224	0,3269	0,3256	0,3200	0,3303	0,3170	0,3291
4c_WDRI_2W	0,2852	0,3025	0,3076	0,3027	0,3150	0,2909	0,2951
4c_WDRI_4W	0,3020	0,3158	0,3059	0,3132	0,3129	0,3213	0,3203
5c_WDRI_1W	0,2503	0,2475	0,2514	0,2421	0,2336	0,2507	0,2716
5c_WDRI_2W	0,2646	0,2329	0,2676	0,2568	0,2873	0,2866	0,2610
5c_WDRI_4W	0,2386	0,2560	0,2445	0,2434	0,2482	0,2352	0,2064

PREDICTED: DOWNDOWN, OBSERVED: DOWN

	min split 50	min split 40	min split 30	min split 25	min split 20	min split 15	min split 10
4c_WDRI_1W	0,3269	0,2969	0,3263	0,3272	0,3271	0,3169	0,3085
4c_WDRI_2W	0,3014	0,2567	0,2325	0,2783	0,2783	0,3086	0,3065
4c_WDRI_4W	0,3620	0,3330	0,3485	0,3294	0,3738	0,3330	0,3313
5c_WDRI_1W	0,3000	0,1899	0,2295	0,2631	0,2460	0,2776	0,2539
5c_WDRI_2W	0,1667	0,2260	0,2051	0,2515	0,2620	0,2217	0,2599
5c_WDRI_4W	0,2124	0,2160	0,2237	0,2183	0,2014	0,2212	0,2241

PREDICTED: DOWN, OBSERVED: DOWNDOWN

	min split 50	min split 40	min split 30	min split 25	min split 20	min split 15	min split 10
4c_WDRI_1W	0,2084	0,2055	0,2041	0,2049	0,2086	0,2114	0,2033
4c_WDRI_2W	0,1993	0,1834	0,1752	0,1858	0,1887	0,1887	0,1947
4c_WDRI_4W	0,2401	0,2447	0,2378	0,2366	0,2428	0,2295	0,2473
5c_WDRI_1W	0,0994	0,1176	0,1176	0,0966	0,1162	0,1088	0,1048
5c_WDRI_2W	0,0974	0,1026	0,0975	0,1007	0,0888	0,0885	0,0968
5c_WDRI_4W	0,2541	0,2524	0,2748	0,2410	0,2387	0,2327	0,2675

PREDICTED: DOWNDOWN, OBSERVED: DOWNDOWN

	min split 50	min split 40	min split 30	min split 25	min split 20	min split 15	min split 10
4c_WDRI_1W	0,2125	0,2212	0,2095	0,2192	0,1966	0,1896	0,1939
4c_WDRI_2W	0,2657	0,2598	0,2794	0,2759	0,2555	0,2494	0,2317
4c_WDRI_4W	0,2195	0,2221	0,2321	0,2486	0,2517	0,2642	0,2314
5c_WDRI_1W	0,1062	0,0886	0,0572	0,0745	0,0947	0,1095	0,1085
5c_WDRI_2W	0,1917	0,0458	0,0709	0,1039	0,1090	0,1003	0,0971
5c_WDRI_4W	0,2584	0,2475	0,2374	0,2376	0,2352	0,2139	0,2097

Appendix E. Portfolio performances

All Sharpe Ratios above the 0.5 level are highlighted

CART standard model

			Yearly re-estimation			Semi-annually re-estimation			Quarterly re-estimation		
			return	*volatility*	*Sharpe Ratio*	*return*	*volatility*	*Sharpe Ratio*	*return*	*volatility*	*Sharpe Ratio*
3 classes	2 weeks	split 10	-0,0126	0,0320	**-0,3937**	0,0024	0,0312	**0,0770**	0,0092	0,0374	**0,2459**
3 classes	2 weeks	split 25	0,0122	0,0412	**0,2962**	0,0204	0,0405	**0,5032**	0,0070	0,0428	**0,1646**
3 classes	2 weeks	split 50	0,0019	0,0441	**0,0424**	0,0146	0,0428	**0,3415**	0,0008	0,0401	**0,0200**
3 classes	4 weeks	split 10	-0,0267	0,0595	**-0,4486**	0,0283	0,0571	**0,4952**	0,0277	0,0496	**0,5579**
3 classes	4 weeks	split 25	-0,0404	0,0603	**-0,6698**	-0,0037	0,0599	**-0,0610**	0,0096	0,0510	**0,1891**
3 classes	4 weeks	split 50	-0,0417	0,0624	**-0,6678**	-0,0373	0,0559	**-0,6670**	-0,0243	0,0514	**-0,4739**
4 classes	2 weeks	split 10	-0,0219	0,0669	**-0,3266**	0,0247	0,0661	**0,3741**	-0,0159	0,0662	**-0,2397**
4 classes	2 weeks	split 25	-0,0149	0,0670	**-0,2225**	0,0205	0,0662	**0,3095**	-0,0005	0,0662	**-0,0071**
4 classes	2 weeks	split 50	-0,0218	0,0669	**-0,3259**	-0,0069	0,0662	**-0,1040**	-0,0397	0,0660	**-0,6012**
4 classes	4 weeks	split 10	0,0129	0,0950	**0,1358**	0,0409	0,0949	**0,4314**	0,0525	0,0948	**0,5542**
4 classes	4 weeks	split 25	-0,0004	0,0950	**-0,0046**	-0,0017	0,0951	**-0,0174**	0,0842	0,0943	**0,8929**
4 classes	4 weeks	split 50	0,0382	0,0949	**0,4025**	0,0206	0,0950	**0,2164**	0,0967	0,0941	**1,0270**
5 classes	2 weeks	split 10	0,0070	0,0585	**0,1191**	0,0068	0,0563	**0,1205**	-0,0159	0,0560	**-0,2841**
5 classes	2 weeks	split 25	0,0094	0,0601	**0,1560**	0,0212	0,0585	**0,3625**	-0,0041	0,0579	**-0,0712**
5 classes	2 weeks	split 50	0,0042	0,0591	**0,0710**	-0,0004	0,0598	**-0,0065**	0,0085	0,0590	**0,1449**
5 classes	4 weeks	split 10	-0,0475	0,0879	**-0,5408**	-0,0345	0,0859	**-0,4015**	-0,0163	0,0883	**-0,1847**
5 classes	4 weeks	split 25	-0,0175	0,0942	**-0,1857**	-0,0131	0,0897	**-0,1460**	0,0108	0,0908	**0,1186**
5 classes	4 weeks	split 50	-0,0298	0,0920	**-0,3233**	-0,0975	0,0859	**-1,1359**	-0,0799	0,0902	**-0,8856**

FACT standard model

			Yearly re-estimation			Semi-annually re-estimation			Quarterly re-estimation		
			return	*volatility*	*Sharpe Ratio*	*return*	*volatility*	*Sharpe Ratio*	*return*	*volatility*	*Sharpe Ratio*
3 classes	2 weeks	split 10	0,0196	0,0577	**0,3401**	0,0505	0,0590	**0,8553**	0,0485	0,0596	**0,8136**
3 classes	2 weeks	split 25	0,0106	0,0550	**0,1919**	0,0341	0,0551	**0,6183**	0,0196	0,0571	**0,3432**
3 classes	2 weeks	split 50	0,0000	0,0629	**-0,0004**	0,0265	0,0631	**0,4204**	0,0226	0,0622	**0,3638**
3 classes	4 weeks	split 10	0,0422	0,0716	**0,5897**	0,0662	0,0704	**0,9408**	0,0654	0,0721	**0,9069**
3 classes	4 weeks	split 25	0,0564	0,0730	**0,7722**	0,0625	0,0729	**0,8577**	0,0165	0,0713	**0,2314**
3 classes	4 weeks	split 50	0,0272	0,0617	**0,4416**	0,0284	0,0632	**0,4496**	-0,0041	0,0682	**-0,0595**
4 classes	2 weeks	split 10	-0,0011	0,0671	**-0,0166**	-0,0178	0,0661	**-0,2687**	-0,0137	0,0661	**-0,2075**
4 classes	2 weeks	split 25	-0,0031	0,0671	**-0,0462**	-0,0325	0,0660	**-0,4931**	0,0104	0,0661	**0,1566**
4 classes	2 weeks	split 50	0,0155	0,0671	**0,2306**	-0,0158	0,0661	**-0,2383**	0,0107	0,0661	**0,1617**
4 classes	4 weeks	split 10	-0,0305	0,0954	**-0,3198**	0,0143	0,0953	**0,1506**	0,0359	0,0952	**0,3769**
4 classes	4 weeks	split 25	-0,0363	0,0954	**-0,3809**	0,0082	0,0953	**0,0861**	0,0532	0,0950	**0,5601**
4 classes	4 weeks	split 50	-0,0995	0,0945	**-1,0525**	-0,0632	0,0949	**-0,6661**	-0,0071	0,0953	**-0,0750**
5 classes	2 weeks	split 10	0,0000	0,0560	**-0,0006**	-0,0076	0,0559	**-0,1364**	-0,0180	0,0557	**-0,3240**
5 classes	2 weeks	split 25	0,0344	0,0515	**0,6686**	0,0252	0,0569	**0,4421**	-0,0252	0,0557	**-0,4537**
5 classes	2 weeks	split 50	0,0046	0,0497	**0,0929**	0,0332	0,0533	**0,6227**	-0,0226	0,0538	**-0,4201**
5 classes	4 weeks	split 10	0,0046	0,0831	**0,0559**	-0,0038	0,0846	**-0,0449**	-0,0055	0,0858	**-0,0641**
5 classes	4 weeks	split 25	-0,0350	0,0859	**-0,4080**	0,0002	0,0876	**0,0020**	-0,0386	0,0875	**-0,4412**
5 classes	4 weeks	split 50	-0,0192	0,0866	**-0,2212**	0,0109	0,0889	**0,1227**	-0,0295	0,0852	**-0,3468**

CRUISE standard model

			Yearly re-estimation			Semi-annually re-estimation			Quarterly re-estimation		
			return	*volatility*	*Sharpe Ratio*	*return*	*volatility*	*Sharpe Ratio*	*return*	*volatility*	*Sharpe Ratio*
3 classes	2 weeks	split 10	-0,0032	0,0567	**-0,0572**	0,0086	0,0528	**0,1631**	0,0160	0,0555	**0,2878**
3 classes	2 weeks	split 25	0,0205	0,0565	**0,3633**	0,0119	0,0551	**0,2155**	0,0015	0,0573	**0,0269**
3 classes	2 weeks	split 50	0,0089	0,0617	**0,1448**	0,0056	0,0593	**0,0951**	0,0055	0,0629	**0,0878**
3 classes	4 weeks	split 10	0,0043	0,0645	**0,0668**	-0,0095	0,0675	**-0,1402**	0,0331	0,0689	**0,4798**
3 classes	4 weeks	split 25	0,0273	0,0694	**0,3941**	0,0040	0,0704	**0,0570**	0,0505	0,0723	**0,6984**
3 classes	4 weeks	split 50	0,0321	0,0741	**0,4325**	0,0110	0,0758	**0,1447**	0,0157	0,0679	**0,2306**
4 classes	2 weeks	split 10	-0,0257	0,0670	**-0,3842**	-0,0406	0,0659	**-0,6160**	-0,0329	0,0660	**-0,4987**
4 classes	2 weeks	split 25	-0,0163	0,0670	**-0,2430**	-0,0360	0,0659	**-0,5462**	0,0138	0,0661	**0,2082**
4 classes	2 weeks	split 50	-0,0336	0,0669	**-0,5028**	-0,0291	0,0660	**-0,4404**	0,0084	0,0661	**0,1267**
4 classes	4 weeks	split 10	0,0529	0,0952	**0,5553**	-0,0042	0,0953	**-0,0439**	-0,0290	0,0952	**-0,3041**
4 classes	4 weeks	split 25	0,0173	0,0955	**0,1813**	0,0000	0,0953	**-0,0004**	-0,0237	0,0952	**-0,2491**
4 classes	4 weeks	split 50	-0,0121	0,0955	**-0,1269**	0,0140	0,0953	**0,1471**	0,0615	0,0949	**0,6478**
5 classes	2 weeks	split 10	-0,0247	0,0448	**-0,5504**	-0,0131	0,0445	**-0,2935**	0,0017	0,0455	**0,0377**
5 classes	2 weeks	split 25	0,0115	0,0475	**0,2426**	0,0089	0,0476	**0,1873**	-0,0018	0,0449	**-0,0402**
5 classes	2 weeks	split 50	0,0250	0,0554	**0,4519**	0,0123	0,0499	**0,2467**	-0,0138	0,0468	**-0,2953**
5 classes	4 weeks	split 10	0,0093	0,0816	**0,1136**	0,0226	0,0805	**0,2803**	0,0177	0,0838	**0,2107**
5 classes	4 weeks	split 25	0,0360	0,0747	**0,4816**	0,0104	0,0821	**0,1270**	-0,0066	0,0848	**-0,0783**
5 classes	4 weeks	split 50	-0,0096	0,0776	**-0,1237**	-0,0535	0,0831	**-0,6444**	-0,0497	0,0861	**-0,5767**

CART with symmetric non-equal misclassification costs

			Yearly re-estimation			Semi-annually re-estimation			Quarterly re-estimation		
			return	*volatility*	*Sharpe Ratio*	*return*	*volatility*	*Sharpe Ratio*	*return*	*volatility*	*Sharpe Ratio*
3 classes	2 weeks	split 10	-0,0071	0,0336	**-0,2126**	0,0118	0,0379	**0,3123**	-0,0170	0,0387	**-0,4382**
3 classes	2 weeks	split 25	-0,0028	0,0375	**-0,0741**	0,0141	0,0401	**0,3509**	-0,0231	0,0372	**-0,6219**
3 classes	2 weeks	split 50	0,0173	0,0350	**0,4939**	0,0165	0,0396	**0,4168**	-0,0205	0,0352	**-0,5838**
3 classes	4 weeks	split 10	-0,0362	0,0472	**-0,7683**	-0,0032	0,0495	**-0,0637**	-0,0045	0,0533	**-0,0845**
3 classes	4 weeks	split 25	-0,0351	0,0550	**-0,6381**	-0,0168	0,0489	**-0,3437**	-0,0013	0,0473	**-0,0273**
3 classes	4 weeks	split 50	-0,0388	0,0553	**-0,7012**	-0,0347	0,0464	**-0,7468**	-0,0142	0,0457	**-0,3112**
4 classes	2 weeks	split 10	-0,0349	0,0668	**-0,5225**	0,0151	0,0662	**0,2278**	-0,0219	0,0662	**-0,3309**
4 classes	2 weeks	split 25	-0,0150	0,0670	**-0,2234**	0,0213	0,0662	**0,3219**	-0,0030	0,0662	**-0,0448**
4 classes	2 weeks	split 50	-0,0299	0,0669	**-0,4467**	0,0034	0,0662	**0,0508**	-0,0207	0,0662	**-0,3122**
4 classes	4 weeks	split 10	0,0129	0,0950	**0,1358**	0,0346	0,0949	**0,3650**	0,0525	0,0948	**0,5542**
4 classes	4 weeks	split 25	-0,0004	0,0950	**-0,0046**	-0,0017	0,0951	**-0,0174**	0,0754	0,0945	**0,7985**
4 classes	4 weeks	split 50	0,0110	0,0950	**0,1162**	-0,0001	0,0951	**-0,0010**	0,0947	0,0941	**1,0054**
5 classes	2 weeks	split 10	-0,0181	0,0523	**-0,3455**	0,0006	0,0518	**0,0112**	-0,0134	0,0522	**-0,2575**
5 classes	2 weeks	split 25	-0,0060	0,0449	**-0,1335**	-0,0012	0,0453	**-0,0267**	-0,0278	0,0474	**-0,5856**
5 classes	2 weeks	split 50	0,0000	0,0346	**-0,0011**	0,0108	0,0340	**0,3177**	-0,0089	0,0377	**-0,2360**
5 classes	4 weeks	split 10	-0,0183	0,0824	**-0,2219**	-0,0168	0,0807	**-0,2082**	0,0229	0,0795	**0,2883**
5 classes	4 weeks	split 25	-0,0300	0,0838	**-0,3579**	-0,0159	0,0793	**-0,2009**	0,0139	0,0764	**0,1818**
5 classes	4 weeks	split 50	-0,0139	0,0784	**-0,1778**	-0,0304	0,0699	**-0,4351**	-0,0232	0,0682	**-0,3395**

FACT with symmetric non-equal misclassification costs

			Yearly re-estimation			Semi-annually re-estimation			Quarterly re-estimation		
			return	*volatility*	*Sharpe Ratio*	*return*	*volatility*	*Sharpe Ratio*	*return*	*volatility*	*Sharpe Ratio*
3 classes	2 weeks	split 10	0,0146	0,0495	**0,2958**	0,0292	0,0472	**0,6192**	0,0285	0,0458	**0,6221**
3 classes	2 weeks	split 25	0,0047	0,0396	**0,1176**	0,0075	0,0337	**0,2220**	0,0038	0,0323	**0,1185**
3 classes	2 weeks	split 50	0,0026	0,0220	**0,1164**	0,0066	0,0171	**0,3863**	0,0064	0,0199	**0,3222**
3 classes	4 weeks	split 10	0,0360	0,0590	**0,6107**	0,0698	0,0606	**1,1510**	0,0341	0,0624	**0,5470**
3 classes	4 weeks	split 25	0,0092	0,0242	**0,3806**	0,0090	0,0309	**0,2907**	0,0034	0,0323	**0,1063**
3 classes	4 weeks	split 50	-0,0094	0,0195	**-0,4845**	-0,0087	0,0287	**-0,3017**	-0,0156	0,0275	**-0,5667**
4 classes	2 weeks	split 10	-0,0107	0,0671	**-0,1594**	-0,0241	0,0660	**-0,3646**	-0,0193	0,0661	**-0,2916**
4 classes	2 weeks	split 25	-0,0063	0,0671	**-0,0941**	-0,0195	0,0661	**-0,2953**	0,0001	0,0661	**0,0022**
4 classes	2 weeks	split 50	-0,0046	0,0671	**-0,0682**	-0,0332	0,0660	**-0,5038**	0,0141	0,0661	**0,2134**
4 classes	4 weeks	split 10	-0,0237	0,0955	**-0,2478**	0,0216	0,0952	**0,2267**	0,0283	0,0952	**0,2976**
4 classes	4 weeks	split 25	-0,0377	0,0954	**-0,3957**	0,0394	0,0951	**0,4139**	0,0459	0,0951	**0,4829**
4 classes	4 weeks	split 50	-0,0628	0,0951	**-0,6605**	0,0069	0,0953	**0,0720**	0,0074	0,0953	**0,0779**
5 classes	2 weeks	split 10	0,0140	0,0387	**0,3615**	0,0051	0,0416	**0,1214**	-0,0036	0,0387	**-0,0930**
5 classes	2 weeks	split 25	0,0055	0,0201	**0,2743**	-0,0040	0,0243	**-0,1645**	-0,0144	0,0259	**-0,5556**
5 classes	2 weeks	split 50	0,0040	0,0159	**0,2489**	0,0014	0,0174	**0,0827**	-0,0120	0,0228	**-0,5233**
5 classes	4 weeks	split 10	-0,0210	0,0555	**-0,3781**	0,0008	0,0626	**0,0132**	0,0165	0,0617	**0,2678**
5 classes	4 weeks	split 25	-0,0437	0,0497	**-0,8791**	-0,0401	0,0514	**-0,7806**	-0,0075	0,0485	**-0,1555**
5 classes	4 weeks	split 50	-0,0181	0,0345	**-0,5231**	-0,0044	0,0369	**-0,1189**	0,0070	0,0311	**0,2268**

CRUISE with symmetric non-equal misclassification costs

			Yearly re-estimation			Semi-annually re-estimation			Quarterly re-estimation		
			return	*volatility*	*Sharpe Ratio*	*return*	*volatility*	*Sharpe Ratio*	*return*	*volatility*	*Sharpe Ratio*
3 classes	2 weeks	split 10	0,0044	0,0298	**0,1493**	0,0171	0,0280	**0,6108**	0,0198	0,0354	**0,5590**
3 classes	2 weeks	split 25	0,0018	0,0238	**0,0770**	0,0139	0,0224	**0,6196**	0,0061	0,0247	**0,2460**
3 classes	2 weeks	split 50	-0,0004	0,0010	**-0,4083**	0,0085	0,0110	**0,7725**	0,0183	0,0177	**1,0295**
3 classes	4 weeks	split 10	0,0437	0,0502	**0,8714**	-0,0323	0,0518	**-0,6233**	0,0239	0,0578	**0,4129**
3 classes	4 weeks	split 25	0,0340	0,0590	**0,5765**	-0,0133	0,0551	**-0,2409**	0,0068	0,0498	**0,1361**
3 classes	4 weeks	split 50	-0,0002	0,0181	**-0,0112**	-0,0196	0,0320	**-0,6147**	-0,0208	0,0249	**-0,8329**
4 classes	2 weeks	split 10	-0,0305	0,0670	**-0,4553**	-0,0482	0,0658	**-0,7323**	-0,0397	0,0659	**-0,6028**
4 classes	2 weeks	split 25	0,0004	0,0671	**0,0065**	-0,0271	0,0660	**-0,4103**	-0,0027	0,0661	**-0,0407**
4 classes	2 weeks	split 50	-0,0487	0,0667	**-0,7301**	-0,0264	0,0660	**-0,3991**	0,0041	0,0661	**0,0620**
4 classes	4 weeks	split 10	0,0383	0,0954	**0,4013**	0,0017	0,0953	**0,0176**	-0,0475	0,0951	**-0,4994**
4 classes	4 weeks	split 25	0,0112	0,0955	**0,1173**	0,0282	0,0952	**0,2962**	-0,0324	0,0952	**-0,3406**
4 classes	4 weeks	split 50	-0,0074	0,0955	**-0,0771**	0,0766	0,0947	**0,8085**	0,0739	0,0947	**0,7806**
5 classes	2 weeks	split 10	-0,0248	0,0259	**-0,9558**	-0,0072	0,0246	**-0,2922**	0,0045	0,0268	**0,1669**
5 classes	2 weeks	split 25	-0,0075	0,0088	**-0,8524**	-0,0034	0,0144	**-0,2360**	0,0026	0,0130	**0,1981**
5 classes	2 weeks	split 50	-0,0017	0,0043	**-0,3835**	0,0001	0,0084	**0,0171**	-0,0017	0,0075	**-0,2202**
5 classes	4 weeks	split 10	0,0536	0,0595	**0,9011**	0,0287	0,0622	**0,4619**	0,0429	0,0586	**0,7329**
5 classes	4 weeks	split 25	0,0390	0,0411	**0,9478**	0,0042	0,0394	**0,1059**	0,0204	0,0440	**0,4626**
5 classes	4 weeks	split 50	0,0183	0,0208	**0,8776**	0,0001	0,0166	**0,0058**	-0,0050	0,0172	**-0,2923**

CART with non-symmetric non-equal misclassification costs

			Yearly re-estimation			Semi-annually re-estimation			Quarterly re-estimation		
			return	*volatility*	*Sharpe Ratio*	*return*	*volatility*	*Sharpe Ratio*	*return*	*volatility*	*Sharpe Ratio*
3 classes	2 weeks	split 10	0,0008	0,0331	**0,0228**	0,0030	0,0307	**0,0984**	-0,0099	0,0374	**-0,2654**
3 classes	2 weeks	split 25	-0,0092	0,0274	**-0,3373**	-0,0102	0,0288	**-0,3561**	-0,0176	0,0292	**-0,6032**
3 classes	2 weeks	split 50	0,0077	0,0300	**0,2563**	-0,0095	0,0295	**-0,3233**	-0,0133	0,0274	**-0,4842**
3 classes	4 weeks	split 10	-0,0515	0,0567	**-0,9079**	0,0023	0,0538	**0,0433**	0,0101	0,0502	**0,2014**
3 classes	4 weeks	split 25	-0,0315	0,0607	**-0,5189**	0,0043	0,0505	**0,0859**	0,0204	0,0468	**0,4370**
3 classes	4 weeks	split 50	-0,0056	0,0522	**-0,1081**	0,0004	0,0400	**0,0091**	0,0067	0,0330	**0,2036**
4 classes	2 weeks	split 10	-0,0149	0,0670	**-0,2232**	0,0194	0,0662	**0,2935**	-0,0090	0,0662	**-0,1353**
4 classes	2 weeks	split 25	0,0014	0,0670	**0,0206**	0,0316	0,0661	**0,4789**	0,0036	0,0662	**0,0551**
4 classes	2 weeks	split 50	-0,0288	0,0669	**-0,4308**	0,0050	0,0662	**0,0762**	-0,0305	0,0661	**-0,4614**
4 classes	4 weeks	split 10	0,0084	0,0950	**0,0885**	0,0250	0,0950	**0,2627**	0,0204	0,0950	**0,2151**
4 classes	4 weeks	split 25	-0,0001	0,0950	**-0,0012**	-0,0063	0,0951	**-0,0664**	0,0571	0,0947	**0,6023**
4 classes	4 weeks	split 50	0,0110	0,0950	**0,1162**	-0,0098	0,0951	**-0,1035**	0,1025	0,0940	**1,0904**
5 classes	2 weeks	split 10	0,0100	0,0501	**0,1995**	-0,0111	0,0546	**-0,2028**	0,0345	0,0539	**0,6414**
5 classes	2 weeks	split 25	-0,0143	0,0504	**-0,2846**	-0,0103	0,0508	**-0,2035**	-0,0053	0,0488	**-0,1091**
5 classes	2 weeks	split 50	-0,0207	0,0478	**-0,4330**	-0,0273	0,0458	**-0,5972**	-0,0002	0,0466	**-0,0034**
5 classes	4 weeks	split 10	-0,0433	0,0851	**-0,5085**	-0,0284	0,0824	**-0,3449**	-0,0336	0,0823	**-0,4087**
5 classes	4 weeks	split 25	-0,0262	0,0833	**-0,3144**	-0,0003	0,0793	**-0,0037**	0,0125	0,0758	**0,1653**
5 classes	4 weeks	split 50	-0,0327	0,0792	**-0,4131**	-0,0111	0,0773	**-0,1432**	-0,0062	0,0772	**-0,0799**

FACT with non-symmetric non-equal misclassification costs

			Yearly re-estimation			Semi-annually re-estimation			Quarterly re-estimation		
			return	*volatility*	*Sharpe Ratio*	*return*	*volatility*	*Sharpe Ratio*	*return*	*volatility*	*Sharpe Ratio*
3 classes	2 weeks	split 10	0,0211	0,0578	**0,3656**	0,0482	0,0584	**0,8252**	0,0517	0,0608	**0,8511**
3 classes	2 weeks	split 25	0,0085	0,0524	**0,1622**	0,0337	0,0537	**0,6281**	0,0287	0,0564	**0,5092**
3 classes	2 weeks	split 50	0,0042	0,0567	**0,0740**	0,0492	0,0578	**0,8506**	0,0351	0,0529	**0,6637**
3 classes	4 weeks	split 10	0,0406	0,0812	**0,4999**	0,0675	0,0821	**0,8224**	0,0592	0,0850	**0,6966**
3 classes	4 weeks	split 25	0,0700	0,0735	**0,9522**	0,0696	0,0790	**0,8810**	0,0399	0,0718	**0,5552**
3 classes	4 weeks	split 50	0,0129	0,0550	**0,2337**	0,0333	0,0686	**0,4850**	0,0215	0,0659	**0,3262**
4 classes	2 weeks	split 10	0,0017	0,0671	**0,0254**	-0,0261	0,0660	**-0,3951**	-0,0056	0,0661	**-0,0845**
4 classes	2 weeks	split 25	-0,0156	0,0670	**-0,2322**	-0,0145	0,0661	**-0,2199**	0,0210	0,0661	**0,3173**
4 classes	2 weeks	split 50	-0,0046	0,0671	**-0,0682**	0,0083	0,0661	**0,1254**	0,0155	0,0661	**0,2345**
4 classes	4 weeks	split 10	-0,0335	0,0954	**-0,3514**	0,0146	0,0953	**0,1533**	0,0127	0,0953	**0,1336**
4 classes	4 weeks	split 25	-0,0449	0,0953	**-0,4708**	0,0199	0,0952	**0,2085**	0,0509	0,0950	**0,5362**
4 classes	4 weeks	split 50	-0,0646	0,0951	**-0,6796**	0,0037	0,0953	**0,0384**	0,0704	0,0948	**0,7422**
5 classes	2 weeks	split 10	0,0018	0,0575	**0,0313**	0,0030	0,0594	**0,0507**	-0,0218	0,0559	**-0,3894**
5 classes	2 weeks	split 25	0,0383	0,0506	**0,7569**	0,0310	0,0567	**0,5467**	-0,0136	0,0554	**-0,2460**
5 classes	2 weeks	split 50	0,0453	0,0513	**0,8830**	0,0463	0,0542	**0,8549**	-0,0021	0,0544	**-0,0384**
5 classes	4 weeks	split 10	-0,0190	0,0868	**-0,2190**	0,0155	0,0868	**0,1783**	0,0336	0,0848	**0,3958**
5 classes	4 weeks	split 25	-0,0684	0,0901	**-0,7596**	0,0013	0,0903	**0,0140**	0,0283	0,0901	**0,3147**
5 classes	4 weeks	split 50	-0,0561	0,0894	**-0,6281**	-0,0016	0,0890	**-0,0179**	0,0140	0,0860	**0,1625**

CRUISE with non-symmetric non-equal misclassification costs

			Yearly re-estimation			Semi-annually re-estimation			Quarterly re-estimation		
			return	*volatility*	*Sharpe Ratio*	*return*	*volatility*	*Sharpe Ratio*	*return*	*volatility*	*Sharpe Ratio*
3 classes	2 weeks	split 10	-0,0075	0,0516	**-0,1451**	0,0117	0,0478	**0,2443**	0,0261	0,0510	**0,5125**
3 classes	2 weeks	split 25	0,0162	0,0531	**0,3052**	0,0118	0,0511	**0,2309**	0,0033	0,0504	**0,0658**
3 classes	2 weeks	split 50	-0,0062	0,0505	**-0,1229**	0,0034	0,0524	**0,0653**	0,0084	0,0535	**0,1569**
3 classes	4 weeks	split 10	0,0094	0,0806	**0,1167**	0,0212	0,0801	**0,2652**	0,0406	0,0795	**0,5104**
3 classes	4 weeks	split 25	0,0215	0,0707	**0,3036**	-0,0086	0,0721	**-0,1191**	0,0341	0,0720	**0,4730**
3 classes	4 weeks	split 50	0,0160	0,0676	**0,2360**	-0,0242	0,0700	**-0,3455**	0,0338	0,0699	**0,4840**
4 classes	2 weeks	split 10	-0,0076	0,0671	**-0,1138**	-0,0316	0,0660	**-0,4795**	-0,0213	0,0661	**-0,3222**
4 classes	2 weeks	split 25	0,0177	0,0670	**0,2643**	-0,0077	0,0661	**-0,1168**	0,0106	0,0661	**0,1605**
4 classes	2 weeks	split 50	0,0050	0,0671	**0,0739**	-0,0110	0,0661	**-0,1664**	0,0031	0,0661	**0,0463**
4 classes	4 weeks	split 10	0,0441	0,0953	**0,4622**	0,0397	0,0951	**0,4169**	-0,0242	0,0952	**-0,2542**
4 classes	4 weeks	split 25	0,0556	0,0952	**0,5837**	0,0554	0,0950	**0,5829**	0,0145	0,0953	**0,1525**
4 classes	4 weeks	split 50	0,0633	0,0951	**0,6653**	0,0908	0,0944	**0,9615**	0,0811	0,0946	**0,8574**
5 classes	2 weeks	split 10	-0,0172	0,0468	**-0,3664**	0,0048	0,0469	**0,1033**	0,0069	0,0483	**0,1431**
5 classes	2 weeks	split 25	0,0122	0,0474	**0,2584**	0,0051	0,0473	**0,1071**	-0,0081	0,0474	**-0,1710**
5 classes	2 weeks	split 50	0,0462	0,0506	**0,9124**	0,0152	0,0497	**0,3051**	0,0089	0,0501	**0,1768**
5 classes	4 weeks	split 10	-0,0244	0,0896	**-0,2721**	0,0110	0,0855	**0,1285**	0,0503	0,0885	**0,5685**
5 classes	4 weeks	split 25	0,0141	0,0821	**0,1716**	0,0305	0,0864	**0,3532**	0,0181	0,0893	**0,2023**
5 classes	4 weeks	split 50	-0,0278	0,0831	**-0,3346**	-0,0431	0,0857	**-0,5031**	-0,0175	0,0879	**-0,1995**

FACT with rolling window

			Yearly re-estimation			Semi-annually re-estimation			Quarterly re-estimation		
			return	*volatility*	*Sharpe Ratio*	*return*	*volatility*	*Sharpe Ratio*	*return*	*volatility*	*Sharpe Ratio*
3 classes	2 weeks	split 10	0,0084	0,0566	**0,1491**	0,0282	0,0544	**0,5178**	0,0160	0,0541	**0,2958**
3 classes	2 weeks	split 25	0,0062	0,0587	**0,1058**	0,0028	0,0578	**0,0487**	0,0113	0,0564	**0,2005**
3 classes	2 weeks	split 50	0,0267	0,0579	**0,4608**	0,0411	0,0590	**0,6972**	0,0152	0,0586	**0,2604**
3 classes	4 weeks	split 10	-0,0562	0,0683	**-0,8218**	-0,0337	0,0653	**-0,5155**	-0,0450	0,0589	**-0,7654**
3 classes	4 weeks	split 25	0,0628	0,0741	**0,8473**	0,0183	0,0691	**0,2645**	-0,0075	0,0657	**-0,1148**
3 classes	4 weeks	split 50	0,0526	0,0773	**0,6811**	0,0170	0,0749	**0,2266**	0,0123	0,0676	**0,1822**
4 classes	2 weeks	split 10	-0,0094	0,0671	**-0,1402**	0,0009	0,0661	**0,0138**	0,0029	0,0661	**0,0440**
4 classes	2 weeks	split 25	0,0001	0,0671	**0,0014**	-0,0207	0,0661	**-0,3129**	-0,0406	0,0659	**-0,6157**
4 classes	2 weeks	split 50	0,0002	0,0671	**0,0027**	-0,0067	0,0661	**-0,1019**	-0,0533	0,0657	**-0,8110**
4 classes	4 weeks	split 10	0,0013	0,0955	**0,0132**	0,0431	0,0951	**0,4529**	-0,0174	0,0953	**-0,1825**
4 classes	4 weeks	split 25	0,0384	0,0954	**0,4032**	0,0297	0,0952	**0,3119**	-0,0148	0,0953	**-0,1558**
4 classes	4 weeks	split 50	0,0516	0,0952	**0,5421**	0,0154	0,0953	**0,1613**	-0,0193	0,0952	**-0,2027**
5 classes	2 weeks	split 10	0,0230	0,0498	**0,4627**	-0,0256	0,0504	**-0,5085**	-0,0288	0,0496	**-0,5812**
5 classes	2 weeks	split 25	0,0033	0,0447	**0,0729**	-0,0081	0,0467	**-0,1730**	-0,0231	0,0475	**-0,4854**
5 classes	2 weeks	split 50	0,0061	0,0362	**0,1699**	-0,0107	0,0409	**-0,2609**	-0,0469	0,0425	**-1,1034**
5 classes	4 weeks	split 10	0,0179	0,0788	**0,2274**	-0,0292	0,0805	**-0,3620**	0,0102	0,0852	**0,1202**
5 classes	4 weeks	split 25	0,0356	0,0823	**0,4319**	-0,0352	0,0828	**-0,4254**	0,0211	0,0835	**0,2532**
5 classes	4 weeks	split 50	0,0464	0,0809	**0,5732**	0,0003	0,0834	**0,0038**	0,0043	0,0823	**0,0528**

FACT with class boundary setting 1

			Yearly re-estimation			Semi-annually re-estimation			Quarterly re-estimation		
			return	*volatility*	*Sharpe Ratio*	*return*	*volatility*	*Sharpe Ratio*	*return*	*volatility*	*Sharpe Ratio*
3 classes	2 weeks	split 10	0,0380	0,0516	**0,7365**	0,0429	0,0528	**0,8113**	0,0725	0,0531	**1,3657**
3 classes	2 weeks	split 15	0,0499	0,0532	**0,9380**	0,0485	0,0544	**0,8904**	0,0662	0,0541	**1,2233**
3 classes	2 weeks	split 20	0,0542	0,0551	**0,9833**	0,0582	0,0557	**1,0454**	0,0680	0,0547	**1,2420**
3 classes	2 weeks	split 25	0,0482	0,0558	**0,8640**	0,0545	0,0561	**0,9712**	0,0647	0,0556	**1,1635**
3 classes	2 weeks	split 35	0,0525	0,0584	**0,8992**	0,0550	0,0569	**0,9667**	0,0768	0,0568	**1,3519**
3 classes	2 weeks	split 50	0,0033	0,0605	**0,0540**	0,0171	0,0582	**0,2944**	0,0531	0,0594	**0,8928**
3 classes	4 weeks	split 10	0,0353	0,0599	**0,5890**	0,0785	0,0542	**1,4478**	0,0853	0,0531	**1,6080**
3 classes	4 weeks	split 15	0,0468	0,0604	**0,7763**	0,0662	0,0539	**1,2298**	0,0768	0,0539	**1,4249**
3 classes	4 weeks	split 20	0,0519	0,0576	**0,9005**	0,0682	0,0519	**1,3133**	0,0703	0,0525	**1,3389**
3 classes	4 weeks	split 25	0,0514	0,0576	**0,8928**	0,0657	0,0526	**1,2503**	0,0670	0,0505	**1,3257**
3 classes	4 weeks	split 35	0,0366	0,0596	**0,6139**	0,0355	0,0571	**0,6218**	0,0642	0,0545	**1,1782**
3 classes	4 weeks	split 50	-0,0011	0,0527	**-0,0211**	-0,0054	0,0514	**-0,1058**	0,0343	0,0518	**0,6625**

FACT with class boundary setting 2

			Yearly re-estimation			Semi-annually re-estimation			Quarterly re-estimation		
			return	*volatility*	*Sharpe Ratio*	*return*	*volatility*	*Sharpe Ratio*	*return*	*volatility*	*Sharpe Ratio*
3 classes	2 weeks	split 10	0,0129	0,0626	**0,2054**	0,0241	0,0624	**0,3858**	0,0258	0,0634	**0,4072**
3 classes	2 weeks	split 15	0,0076	0,0651	**0,1171**	0,0073	0,0637	**0,1140**	0,0013	0,0641	**0,0202**
3 classes	2 weeks	split 20	-0,0031	0,0657	**-0,0476**	0,0140	0,0643	**0,2170**	0,0181	0,0644	**0,2809**
3 classes	2 weeks	split 25	-0,0063	0,0657	**-0,0952**	0,0117	0,0644	**0,1823**	0,0164	0,0644	**0,2551**
3 classes	2 weeks	split 35	-0,0225	0,0656	**-0,3425**	-0,0068	0,0648	**-0,1045**	-0,0023	0,0650	**-0,0351**
3 classes	2 weeks	split 50	-0,0487	0,0662	**-0,7354**	-0,0345	0,0655	**-0,5270**	-0,0485	0,0654	**-0,7427**
3 classes	4 weeks	split 10	0,0266	0,0590	**0,4505**	0,0350	0,0554	**0,6321**	0,0547	0,0569	**0,9619**
3 classes	4 weeks	split 15	0,0325	0,0597	**0,5445**	0,0423	0,0557	**0,7589**	0,0600	0,0575	**1,0448**
3 classes	4 weeks	split 20	0,0291	0,0596	**0,4877**	0,0481	0,0555	**0,8653**	0,0533	0,0550	**0,9682**
3 classes	4 weeks	split 25	0,0296	0,0607	**0,4882**	0,0619	0,0557	**1,1109**	0,0338	0,0550	**0,6151**
3 classes	4 weeks	split 35	0,0122	0,0629	**0,1936**	0,0577	0,0581	**0,9933**	0,0483	0,0574	**0,8416**
3 classes	4 weeks	split 50	0,0066	0,0620	**0,1058**	0,0366	0,0575	**0,6360**	0,0196	0,0573	**0,3417**

FACT with class boundary setting 3

			Yearly re-estimation			Semi-annually re-estimation			Quarterly re-estimation		
			return	*volatility*	*Sharpe Ratio*	*return*	*volatility*	*Sharpe Ratio*	*return*	*volatility*	*Sharpe Ratio*
3 classes	2 weeks	split 10	0,0175	0,0504	**0,3467**	-0,0024	0,0503	**-0,0473**	-0,0049	0,0481	**-0,1016**
3 classes	2 weeks	split 15	0,0191	0,0515	**0,3706**	0,0036	0,0516	**0,0704**	0,0052	0,0501	**0,1037**
3 classes	2 weeks	split 20	0,0130	0,0481	**0,2695**	-0,0051	0,0500	**-0,1027**	0,0075	0,0504	**0,1494**
3 classes	2 weeks	split 25	0,0074	0,0492	**0,1495**	-0,0091	0,0506	**-0,1797**	0,0045	0,0513	**0,0869**
3 classes	2 weeks	split 35	-0,0016	0,0509	**-0,0322**	0,0066	0,0516	**0,1287**	0,0241	0,0507	**0,4755**
3 classes	2 weeks	split 50	0,0001	0,0566	**0,0019**	0,0121	0,0545	**0,2219**	0,0084	0,0521	**0,1604**
3 classes	4 weeks	split 10	-0,0033	0,0494	**-0,0662**	-0,0052	0,0498	**-0,1052**	0,0038	0,0520	**0,0740**
3 classes	4 weeks	split 15	-0,0065	0,0479	**-0,1368**	-0,0051	0,0492	**-0,1031**	-0,0002	0,0510	**-0,0039**
3 classes	4 weeks	split 20	-0,0052	0,0508	**-0,1021**	-0,0132	0,0519	**-0,2534**	0,0015	0,0521	**0,0287**
3 classes	4 weeks	split 25	-0,0127	0,0520	**-0,2437**	-0,0173	0,0509	**-0,3406**	0,0011	0,0519	**0,0216**
3 classes	4 weeks	split 35	-0,0213	0,0523	**-0,4081**	-0,0096	0,0538	**-0,1793**	-0,0023	0,0525	**-0,0443**
3 classes	4 weeks	split 50	-0,0065	0,0544	**-0,1198**	0,0010	0,0548	**0,0178**	0,0065	0,0521	**0,1244**

Appendix F. Out-of-sample investment decisions and performance

3 classes and 2 weeks time horizon

splitmin 10

Q3 2007 prediction	Q3 2007 payoff	Q4 2007 prediction	Q4 2007 payoff	jan-08 prediction	jan-08 payoff
sell	-2,77%	sell	-1,30%	hold	0,00%
sell	-2,15%	buy	2,63%	sell	1,72%
sell	-1,26%	sell	-1,00%	sell	0,10%
sell	-0,61%	buy	1,25%	sell	-0,24%
hold	0,00%	buy	0,45%	sell	0,24%
sell	-1,55%	sell	0,52%		**1,82%**
sell	-0,64%	sell	1,34%		
hold	0,00%	sell	1,94%	**Tot. Ret**	**4,35%**
hold	0,00%	sell	1,11%	**Volatility**	**1,35%**
buy	-1,79%	buy	0,70%		
sell	2,02%	hold	0,00%		
buy	0,79%	hold	0,00%		
hold	0,00%	buy	2,85%		
	-7,96%		**10,49%**		

splitmin 15

Q3 2007 prediction	Q3 2007 payoff	Q4 2007 prediction	Q4 2007 payoff	jan-08 prediction	jan-08 payoff
sell	-2,77%	sell	-1,30%	hold	0,00%
sell	-2,15%	buy	2,63%	sell	1,72%
sell	-1,26%	sell	-1,00%	sell	0,10%
sell	-0,61%	buy	1,25%	sell	-0,24%
hold	0,00%	buy	0,45%	sell	0,24%
sell	-1,55%	sell	0,52%		**1,82%**
sell	-0,64%	sell	1,34%		
hold	0,00%	buy	-1,94%	**Tot. Ret**	**-4,72%**
hold	0,00%	buy	-1,11%	**Volatility**	**1,45%**
buy	-1,79%	buy	0,70%		
sell	2,02%	hold	0,00%		
buy	0,79%	sell	-2,97%		
hold	0,00%	buy	2,85%		
	-7,96%		**1,43%**		

splitmin 20

Q3 2007 prediction	Q3 2007 payoff	Q4 2007 prediction	Q4 2007 payoff	jan-08 prediction	jan-08 payoff
sell	-2,77%	sell	-1,30%	hold	0,00%
sell	-2,15%	buy	2,63%	sell	1,72%
sell	-1,26%	sell	-1,00%	sell	0,10%
sell	-0,61%	buy	1,25%	sell	-0,24%
hold	0,00%	buy	0,45%	sell	0,24%
sell	-1,55%	sell	0,52%		**1,82%**
sell	-0,64%	sell	1,34%		
hold	0,00%	buy	-1,94%	**Tot. Ret**	**-3,76%**
sell	0,96%	buy	-1,11%	**Volatility**	**1,46%**
buy	-1,79%	buy	0,70%		
sell	2,02%	hold	0,00%		
buy	0,79%	sell	-2,97%		
hold	0,00%	buy	2,85%		
	-7,01%		**1,43%**		

splitmin 25

Q3 2007 prediction	Q3 2007 payoff	Q4 2007 prediction	Q4 2007 payoff
sell	-2,77%	sell	-1,30%
sell	-2,15%	buy	2,63%
sell	-1,26%	hold	0,00%
sell	-0,61%	buy	1,25%
sell	-0,72%	buy	0,45%
sell	-1,55%	sell	0,52%
sell	-0,64%	sell	1,34%
sell	0,95%	buy	-1,94%
sell	0,96%	buy	-1,11%
sell	1,79%	buy	0,70%
sell	2,02%	hold	0,00%
buy	0,79%	sell	-2,97%
sell	-0,96%	buy	2,85%
	-4,15%		**2,42%**

jan-08 prediction	jan-08 payoff
hold	0,00%
sell	1,72%
sell	0,10%
sell	-0,24%
sell	0,24%
	1,82%

Tot. Ret	**0,09%**
Volatility	**1,48%**

splitmin 35

Q3 2007 prediction	Q3 2007 payoff	Q4 2007 prediction	Q4 2007 payoff
sell	-2,77%	sell	-1,30%
sell	-2,15%	buy	2,63%
sell	-1,26%	hold	0,00%
sell	-0,61%	buy	1,25%
sell	-0,72%	buy	0,45%
sell	-1,55%	sell	0,52%
sell	-0,64%	sell	1,34%
sell	0,95%	buy	-1,94%
sell	0,96%	buy	-1,11%
sell	1,79%	buy	0,70%
sell	2,02%	buy	2,01%
buy	0,79%	sell	-2,97%
sell	-0,96%	buy	2,85%
	-4,15%		**4,43%**

jan-08 prediction	jan-08 payoff
hold	0,00%
sell	1,72%
sell	0,10%
sell	-0,24%
sell	0,24%
	1,82%

Tot. Ret	**2,10%**
Volatility	**1,53%**

splitmin 50

Q3 2007 prediction	Q3 2007 payoff	Q4 2007 prediction	Q4 2007 payoff
sell	-2,77%	sell	-1,30%
sell	-2,15%	buy	2,63%
sell	-1,26%	hold	0,00%
sell	-0,61%	hold	0,00%
sell	-0,72%	hold	0,00%
sell	-1,55%	sell	0,52%
sell	-0,64%	sell	1,34%
sell	0,95%	buy	-1,94%
sell	0,96%	buy	-1,11%
sell	1,79%	buy	0,70%
sell	2,02%	buy	2,01%
sell	-0,79%	sell	-2,97%
sell	-0,96%	buy	2,85%
	-5,73%		**2,74%**

jan-08 prediction	jan-08 payoff
hold	0,00%
sell	1,72%
sell	0,10%
sell	-0,24%
sell	0,24%
	1,82%

Tot. Ret	**-1,18%**
Volatility	**1,51%**

3 classes and 2 weeks time horizon

splitmin 10

2nd half 2007	
prediction	*payoff*
sell	-2,77%
sell	-2,15%
sell	-1,26%
sell	-0,61%
hold	0,00%
sell	-1,55%
sell	-0,64%
hold	0,00%
hold	0,00%
buy	-1,79%
sell	2,02%
buy	0,79%
hold	0,00%
buy	1,30%
hold	0,00%
sell	-1,00%
sell	-1,25%
hold	0,00%
sell	0,52%
buy	-1,34%
buy	-1,94%
hold	0,00%
buy	0,70%
hold	0,00%
sell	-2,97%
buy	2,85%
	-11,10%

jan-08	
prediction	*payoff*
hold	0,00%
buy	1,72%
buy	0,10%
buy	-0,24%
buy	0,24%
	1,82%

Tot. Ret	**-9,28%**
Volatility	**1,33%**

splitmin 25

2nd half 2007	
prediction	*payoff*
c	-2,77%
sell	-2,15%
sell	-1,26%
sell	-0,61%
sell	-0,72%
sell	-1,55%
sell	-0,64%
sell	0,95%
sell	0,96%
sell	1,79%
sell	2,02%
buy	0,79%
sell	-0,96%
sell	-1,30%
sell	-2,63%
sell	-1,00%
sell	-1,25%
sell	-0,45%
sell	0,52%
sell	1,34%
sell	1,94%
sell	1,11%
buy	0,70%
hold	0,00%
sell	-2,97%
buy	2,85%
	-5,28%

jan-08	
prediction	*payoff*
hold	0,00%
buy	1,72%
buy	0,10%
buy	-0,24%
buy	0,24%
	1,82%

Tot. Ret	**-3,46%**
Volatility	**1,49%**

splitmin 15

2nd half 2007	
prediction	*payoff*
sell	-2,77%
sell	-2,15%
sell	-1,26%
sell	-0,61%
hold	0,00%
sell	-1,55%
sell	-0,64%
hold	0,00%
hold	0,00%
buy	-1,79%
sell	2,02%
buy	0,79%
hold	0,00%
buy	1,30%
hold	0,00%
sell	-1,00%
sell	-1,25%
hold	0,00%
sell	0,52%
buy	-1,34%
buy	-1,94%
hold	0,00%
buy	0,70%
hold	0,00%
sell	-2,97%
buy	2,85%
	-11,10%

jan-08	
prediction	*payoff*
hold	0,00%
buy	1,72%
buy	0,10%
buy	-0,24%
buy	0,24%
	1,82%

Tot. Ret	**-9,28%**
Volatility	**1,33%**

splitmin 35

2nd half 2007	
prediction	*payoff*
sell	-2,77%
sell	-2,15%
sell	-1,26%
sell	-0,61%
sell	-0,72%
sell	-1,55%
sell	-0,64%
sell	0,95%
sell	0,96%
sell	1,79%
sell	2,02%
buy	0,79%
sell	-0,96%
sell	-1,30%
sell	-2,63%
sell	-1,00%
sell	-1,25%
sell	-0,45%
sell	0,52%
sell	1,34%
sell	1,94%
sell	1,11%
buy	0,70%
hold	0,00%
sell	-2,97%
buy	2,85%
	-5,28%

jan-08	
prediction	*payoff*
hold	0,00%
buy	1,72%
buy	0,10%
buy	-0,24%
buy	0,24%
	1,82%

Tot. Ret	**-3,46%**
Volatility	**1,49%**

splitmin 20

2nd half 2007	
prediction	*payoff*
sell	-2,77%
sell	-2,15%
sell	-1,26%
sell	-0,61%
hold	0,00%
sell	-1,55%
sell	-0,64%
hold	0,00%
sell	0,96%
buy	-1,79%
sell	2,02%
buy	0,79%
hold	0,00%
sell	-1,30%
hold	0,00%
sell	-1,00%
sell	-1,25%
hold	0,00%
sell	0,52%
sell	1,34%
sell	1,94%
hold	0,00%
buy	0,70%
hold	0,00%
sell	-2,97%
buy	2,85%
	-6,17%

jan-08	
prediction	*payoff*
hold	0,00%
buy	1,72%
buy	0,10%
buy	-0,24%
buy	0,24%
	1,82%

Tot. Ret	**-4,35%**
Volatility	**1,36%**

splitmin 50

2nd half 2007	
prediction	*payoff*
sell	-2,77%
sell	-2,15%
sell	-1,26%
sell	-0,61%
sell	-0,72%
sell	-1,55%
sell	-0,64%
sell	0,95%
sell	0,96%
sell	1,79%
sell	2,02%
sell	-0,79%
sell	-0,96%
sell	-1,30%
sell	-2,63%
sell	-1,00%
sell	-1,25%
sell	-0,45%
sell	0,52%
sell	1,34%
sell	1,94%
sell	1,11%
sell	-0,70%
sell	-2,01%
sell	-2,97%
sell	-2,85%
	-15,97%

jan-08	
prediction	*payoff*
hold	0,00%
buy	1,72%
buy	0,10%
buy	-0,24%
buy	0,24%
	1,82%

Tot. Ret	**-14,15%**
Volatility	**1,47%**

3 classes and 2 weeks time horizon

	2nd half hold007		
splitmin 10	*prediction*	*payoff*	
	sell	-2,77%	
	sell	-2,15%	
	sell	-1,26%	
	sell	-0,61%	
	hold	0,00%	
	sell	-1,55%	
	sell	-0,64%	
	hold	0,00%	
	hold	0,00%	
	buy	-1,79%	
	sell	2,02%	
	buy	0,79%	
	hold	0,00%	
	buy	1,30%	
	hold	0,00%	
	sell	-1,00%	
	sell	-1,25%	
	hold	0,00%	
	sell	0,52%	
	buy	-1,34%	
	buy	-1,94%	
	hold	0,00%	
	buy	0,70%	
	hold	0,00%	
	sell	-2,97%	
	buy	2,85%	
	buy	1,64%	
	buy	1,72%	
	sell	-0,10%	
	hold	0,00%	
	sell	-0,24%	**volatility**
		-8,08%	**1,37%**

	2nd half hold007		
splitmin 25	*prediction*	*payoff*	
	sell	-2,77%	
	sell	-2,15%	
	sell	-1,26%	
	sell	-0,61%	
	sell	-0,72%	
	sell	-1,55%	
	sell	-0,64%	
	sell	0,95%	
	sell	0,96%	
	sell	1,79%	
	sell	2,02%	
	buy	0,79%	
	sell	-0,96%	
	sell	-1,30%	
	sell	-2,63%	
	sell	-1,00%	
	sell	-1,25%	
	sell	-0,45%	
	sell	0,52%	
	sell	1,34%	
	sell	1,94%	
	sell	1,11%	
	buy	0,70%	
	hold	0,00%	
	sell	-2,97%	
	buy	2,85%	
	buy	1,64%	
	sell	-1,72%	
	sell	-0,10%	
	sell	0,24%	
	sell	-0,24%	**volatility**
		-5,47%	**1,52%**

	2nd half hold007		
splitmin 15	*prediction*	*payoff*	
	sell	-2,77%	
	sell	-2,15%	
	sell	-1,26%	
	sell	-0,61%	
	hold	0,00%	
	sell	-1,55%	
	sell	-0,64%	
	hold	0,00%	
	hold	0,00%	
	buy	-1,79%	
	sell	2,02%	
	buy	0,79%	
	hold	0,00%	
	buy	1,30%	
	hold	0,00%	
	sell	-1,00%	
	sell	-1,25%	
	hold	0,00%	
	sell	0,52%	
	buy	-1,34%	
	buy	-1,94%	
	hold	0,00%	
	buy	0,70%	
	hold	0,00%	
	sell	-2,97%	
	buy	2,85%	
	buy	1,64%	
	buy	1,72%	
	sell	-0,10%	
	hold	0,00%	
	sell	-0,24%	**volatility**
		-8,08%	**1,37%**

	2nd half hold007		
splitmin 35	*prediction*	*payoff*	
	sell	-2,77%	
	sell	-2,15%	
	sell	-1,26%	
	sell	-0,61%	
	sell	-0,72%	
	sell	-1,55%	
	sell	-0,64%	
	sell	0,95%	
	sell	0,96%	
	sell	1,79%	
	sell	2,02%	
	buy	0,79%	
	sell	-0,96%	
	sell	-1,30%	
	sell	-2,63%	
	sell	-1,00%	
	sell	-1,25%	
	sell	-0,45%	
	sell	0,52%	
	sell	1,34%	
	sell	1,94%	
	sell	1,11%	
	buy	0,70%	
	hold	0,00%	
	sell	-2,97%	
	buy	2,85%	
	buy	1,64%	
	sell	-1,72%	
	sell	-0,10%	
	sell	0,24%	
	sell	-0,24%	**volatility**
		-5,47%	**1,52%**

	2nd half hold007		
splitmin 20	*prediction*	*payoff*	
	sell	-2,77%	
	sell	-2,15%	
	sell	-1,26%	
	sell	-0,61%	
	hold	0,00%	
	sell	-1,55%	
	sell	-0,64%	
	hold	0,00%	
	sell	0,96%	
	buy	-1,79%	
	sell	2,02%	
	buy	0,79%	
	hold	0,00%	
	sell	-1,30%	
	hold	0,00%	
	sell	-1,00%	
	sell	-1,25%	
	hold	0,00%	
	sell	0,52%	
	sell	1,34%	
	sell	1,94%	
	hold	0,00%	
	buy	0,70%	
	hold	0,00%	
	sell	-2,97%	
	buy	2,85%	
	buy	1,64%	
	sell	-1,72%	
	sell	-0,10%	
	hold	0,00%	
	sell	-0,24%	**volatility**
		-6,60%	**1,39%**

	2nd half hold007		
splitmin 50	*prediction*	*payoff*	
	sell	-2,77%	
	sell	-2,15%	
	sell	-1,26%	
	sell	-0,61%	
	sell	-0,72%	
	sell	-1,55%	
	sell	-0,64%	
	sell	0,95%	
	sell	0,96%	
	sell	1,79%	
	sell	2,02%	
	sell	-0,79%	
	sell	-0,96%	
	sell	-1,30%	
	sell	-2,63%	
	sell	-1,00%	
	sell	-1,25%	
	sell	-0,45%	
	sell	0,52%	
	sell	1,34%	
	sell	1,94%	
	sell	1,11%	
	sell	-0,70%	
	sell	-2,01%	
	sell	-2,97%	
	sell	-2,85%	
	sell	-1,64%	
	sell	-1,72%	
	sell	-0,10%	
	sell	0,24%	
	sell	-0,24%	**volatility**
		-19,43%	**1,43%**

3 classes and 4 weeks time horizon

splitmin 10

Q3 2007		Q4 2007		jan-08	
prediction	*payoff*	*prediction*	*payoff*	*prediction*	*payoff*
hold	0,00%	buy	1,30%	buy	1,64%
hold	0,00%	buy	2,63%	buy	1,72%
buy	1,26%	sell	-1,00%	buy	0,10%
sell	-0,61%	buy	1,25%	buy	-0,24%
sell	-0,72%	buy	0,45%	buy	0,24%
sell	-1,55%	hold	0,00%		**3,46%**
sell	-0,64%	hold	0,00%		
sell	0,95%	hold	0,00%	**Tot. Ret**	**11,80%**
sell	0,96%	hold	0,00%	**Volatility**	**1,16%**
sell	1,79%	sell	-0,70%		
sell	2,02%	sell	-2,01%		
hold	0,00%	buy	2,97%		
hold	0,00%	hold	0,00%		
	3,45%		**4,89%**		

splitmin 15

Q3 2007		Q4 2007		jan-08	
prediction	*payoff*	*prediction*	*payoff*	*prediction*	*payoff*
hold	0,00%	buy	1,30%	buy	1,64%
hold	0,00%	buy	2,63%	buy	1,72%
buy	1,26%	sell	-1,00%	buy	0,10%
sell	-0,61%	buy	1,25%	buy	-0,24%
sell	-0,72%	buy	0,45%	buy	0,24%
sell	-1,55%	hold	0,00%		**3,46%**
sell	-0,64%	hold	0,00%		
sell	0,95%	hold	0,00%	**Tot. Ret**	**11,80%**
sell	0,96%	hold	0,00%	**Volatility**	**1,16%**
sell	1,79%	sell	-0,70%		
sell	2,02%	sell	-2,01%		
hold	0,00%	buy	2,97%		
hold	0,00%	hold	0,00%		
	3,45%		**4,89%**		

splitmin 20

Q3 2007		Q4 2007		jan-08	
prediction	*payoff*	*prediction*	*payoff*	*prediction*	*payoff*
hold	0,00%	buy	1,30%	buy	1,64%
hold	0,00%	buy	2,63%	buy	1,72%
sell	-1,26%	sell	-1,00%	buy	0,10%
sell	-0,61%	buy	1,25%	buy	-0,24%
sell	-0,72%	buy	0,45%	buy	0,24%
sell	-1,55%	hold	0,00%		**3,46%**
sell	-0,64%	hold	0,00%		
sell	0,95%	hold	0,00%	**Tot. Ret**	**7,53%**
sell	0,96%	hold	0,00%	**Volatility**	**1,22%**
sell	1,79%	sell	-0,70%		
sell	2,02%	sell	-2,01%		
sell	-0,79%	buy	2,97%		
sell	-0,96%	hold	0,00%		
	-0,82%		**4,89%**		

splitmin 25

Q3 2007	
prediction	*payoff*
hold	0,00%
hold	0,00%
sell	-1,26%
sell	-0,61%
sell	-0,72%
sell	-1,55%
sell	-0,64%
sell	0,95%
sell	0,96%
sell	1,79%
sell	2,02%
sell	-0,79%
sell	-0,96%
	-0,82%

Q4 2007	
prediction	*payoff*
buy	1,30%
buy	2,63%
sell	-1,00%
buy	1,25%
buy	0,45%
hold	0,00%
hold	0,00%
hold	0,00%
hold	0,00%
sell	-0,70%
sell	-2,01%
buy	2,97%
hold	0,00%
	4,89%

jan-08	
prediction	*payoff*
buy	1,64%
buy	1,72%
buy	0,10%
buy	-0,24%
buy	0,24%
	3,46%

Tot. Ret	**7,53%**
Volatility	**1,22%**

splitmin 35

Q3 2007	
prediction	*payoff*
hold	0,00%
hold	0,00%
sell	-1,26%
sell	-0,61%
sell	-0,72%
sell	-1,55%
sell	-0,64%
sell	0,95%
sell	0,96%
sell	1,79%
sell	2,02%
sell	-0,79%
sell	-0,96%
	-0,82%

Q4 2007	
prediction	*payoff*
buy	1,30%
buy	2,63%
sell	-1,00%
buy	1,25%
buy	0,45%
hold	0,00%
hold	0,00%
hold	0,00%
hold	0,00%
sell	-0,70%
sell	-2,01%
buy	2,97%
hold	0,00%
	4,89%

jan-08	
prediction	*payoff*
buy	1,64%
buy	1,72%
buy	0,10%
buy	-0,24%
buy	0,24%
	3,46%

Tot. Ret	**7,53%**
Volatility	**1,22%**

splitmin 50

Q3 2007	
prediction	*payoff*
hold	0,00%
hold	0,00%
sell	-1,26%
sell	-0,61%
sell	-0,72%
sell	-1,55%
sell	-0,64%
sell	0,95%
sell	0,96%
sell	1,79%
sell	2,02%
sell	-0,79%
sell	-0,96%
	-0,82%

Q4 2007	
prediction	*payoff*
sell	-1,30%
sell	-2,63%
sell	-1,00%
buy	1,25%
buy	0,45%
hold	0,00%
hold	0,00%
hold	0,00%
hold	0,00%
sell	-0,70%
sell	-2,01%
buy	2,97%
hold	0,00%
	-2,97%

jan-08	
prediction	*payoff*
buy	1,64%
buy	1,72%
buy	0,10%
buy	-0,24%
buy	0,24%
	3,46%

Tot. Ret	**-0,33%**
Volatility	**1,24%**

3 classes and 4 weeks time horizon

splitmin 10

2nd half 2007

prediction	*payoff*
hold	0,00%
hold	0,00%
buy	1,26%
sell	-0,61%
sell	-0,72%
sell	-1,55%
sell	-0,64%
sell	0,95%
sell	0,96%
sell	1,79%
sell	2,02%
hold	0,00%
hold	0,00%
buy	1,30%
hold	0,00%
sell	-1,00%
hold	0,00%
sell	-0,45%
sell	0,52%
sell	1,34%
sell	1,94%
sell	1,11%
sell	-0,70%
sell	-2,01%
hold	0,00%
sell	-2,85%
	2,66%

jan 208

prediction	*payoff*
buy	1,64%
buy	1,72%
buy	0,10%
buy	-0,24%
buy	0,24%
	3,46%

Tot. Ret	**6,12%**
Volatility	**1,18%**

splitmin 25

2nd half 2007

prediction	*payoff*
hold	0,00%
hold	0,00%
sell	-1,26%
sell	-0,61%
sell	-0,72%
sell	-1,55%
sell	-0,64%
sell	0,95%
sell	0,96%
sell	1,79%
sell	2,02%
sell	-0,79%
sell	-0,96%
sell	-1,30%
sell	-2,63%
sell	-1,00%
sell	-1,25%
sell	-0,45%
sell	0,52%
sell	1,34%
sell	1,94%
sell	1,11%
sell	-0,70%
sell	-2,01%
sell	-2,97%
sell	-2,85%
	-11,06%

jan 208

prediction	*payoff*
buy	1,64%
buy	1,72%
buy	0,10%
buy	-0,24%
buy	0,24%
	3,46%

Tot. Ret	**-7,60%**
Volatility	**1,41%**

splitmin 15

2nd half 2007

prediction	*payoff*
hold	0,00%
hold	0,00%
buy	1,26%
sell	-0,61%
sell	-0,72%
sell	-1,55%
sell	-0,64%
sell	0,95%
sell	0,96%
sell	1,79%
sell	2,02%
hold	0,00%
hold	0,00%
buy	1,30%
hold	0,00%
sell	-1,00%
hold	0,00%
sell	-0,45%
sell	0,52%
sell	1,34%
sell	1,94%
sell	1,11%
sell	-0,70%
sell	-2,01%
hold	0,00%
sell	-2,85%
	2,66%

jan 208

prediction	*payoff*
buy	1,64%
buy	1,72%
buy	0,10%
buy	-0,24%
buy	0,24%
	3,46%

Tot. Ret	**6,12%**
Volatility	**1,18%**

splitmin 35

2nd half 2007

prediction	*payoff*
hold	0,00%
hold	0,00%
sell	-1,26%
sell	-0,61%
sell	-0,72%
sell	-1,55%
sell	-0,64%
sell	0,95%
sell	0,96%
sell	1,79%
sell	2,02%
sell	-0,79%
sell	-0,96%
sell	-1,30%
sell	-2,63%
sell	-1,00%
sell	-1,25%
sell	-0,45%
sell	0,52%
sell	1,34%
sell	1,94%
sell	1,11%
sell	-0,70%
sell	-2,01%
sell	-2,97%
sell	-2,85%
	-11,06%

jan 208

prediction	*payoff*
buy	1,64%
buy	1,72%
buy	0,10%
buy	-0,24%
buy	0,24%
	3,46%

Tot. Ret	**-7,60%**
Volatility	**1,41%**

splitmin 20

2nd half 2007

prediction	*payoff*
hold	0,00%
hold	0,00%
sell	-1,26%
sell	-0,61%
sell	-0,72%
sell	-1,55%
sell	-0,64%
sell	0,95%
sell	0,96%
sell	1,79%
sell	2,02%
sell	-0,79%
sell	-0,96%
sell	-1,30%
sell	-2,63%
sell	-1,00%
sell	-1,25%
sell	-0,45%
sell	0,52%
sell	1,34%
sell	1,94%
sell	1,11%
sell	-0,70%
sell	-2,01%
sell	-2,97%
sell	-2,85%
	-11,06%

jan 208

prediction	*payoff*
buy	1,64%
buy	1,72%
buy	0,10%
buy	-0,24%
buy	0,24%
	3,46%

Tot. Ret	**-7,60%**
Volatility	**1,41%**

splitmin 50

2nd half 2007

prediction	*payoff*
hold	0,00%
hold	0,00%
sell	-1,26%
sell	-0,61%
sell	-0,72%
sell	-1,55%
sell	-0,64%
sell	0,95%
sell	0,96%
sell	1,79%
sell	2,02%
sell	-0,79%
sell	-0,96%
sell	-1,30%
sell	-2,63%
sell	-1,00%
sell	-1,25%
sell	-0,45%
sell	0,52%
sell	1,34%
sell	1,94%
sell	1,11%
sell	-0,70%
sell	-2,01%
sell	-2,97%
sell	-2,85%
	-11,06%

jan 208

prediction	*payoff*
buy	1,64%
buy	1,72%
buy	0,10%
buy	-0,24%
buy	0,24%
	3,46%

Tot. Ret	**-7,60%**
Volatility	**1,41%**

3 classes and 4 weeks time horizon

	2nd half 2007		
splitmin 10	*prediction*	*payoff*	
	hold	0,00%	
	hold	0,00%	
	buy	1,26%	
	sell	-0,61%	
	sell	-0,72%	
	sell	-1,55%	
	sell	-0,64%	
	sell	0,95%	
	sell	0,96%	
	sell	1,79%	
	sell	2,02%	
	hold	0,00%	
	hold	0,00%	
	buy	1,30%	
	hold	0,00%	
	sell	-1,00%	
	hold	0,00%	
	sell	-0,45%	
	sell	0,52%	
	sell	1,34%	
	sell	1,94%	
	sell	1,11%	
	sell	-0,70%	
	sell	-2,01%	
	hold	0,00%	
	sell	-2,85%	
	sell	-1,64%	
	sell	-1,72%	
	sell	-0,10%	
	sell	0,24%	
	sell	-0,24%	**volatility**
		-0,79%	**1,19%**

	2nd half 2007		
splitmin 25	*prediction*	*payoff*	
	hold	0,00%	
	hold	0,00%	
	sell	-1,26%	
	sell	-0,61%	
	sell	-0,72%	
	sell	-1,55%	
	sell	-0,64%	
	sell	0,95%	
	sell	0,96%	
	sell	1,79%	
	sell	2,02%	
	sell	-0,79%	
	sell	-0,96%	
	sell	-1,30%	
	sell	-2,63%	
	sell	-1,00%	
	sell	-1,25%	
	sell	-0,45%	
	sell	0,52%	
	sell	1,34%	
	sell	1,94%	
	sell	1,11%	
	sell	-0,70%	
	sell	-2,01%	
	sell	-2,97%	
	sell	-2,85%	
	sell	-1,64%	
	sell	-1,72%	
	sell	-0,10%	
	sell	0,24%	
	sell	-0,24%	**volatility**
		-14,51%	**1,35%**

	2nd half 2007		
splitmin 15	*prediction*	*payoff*	
	hold	0,00%	
	hold	0,00%	
	buy	1,26%	
	sell	-0,61%	
	sell	-0,72%	
	sell	-1,55%	
	sell	-0,64%	
	sell	0,95%	
	sell	0,96%	
	sell	1,79%	
	sell	2,02%	
	hold	0,00%	
	hold	0,00%	
	buy	1,30%	
	hold	0,00%	
	sell	-1,00%	
	hold	0,00%	
	sell	-0,45%	
	sell	0,52%	
	sell	1,34%	
	sell	1,94%	
	sell	1,11%	
	sell	-0,70%	
	sell	-2,01%	
	hold	0,00%	
	sell	-2,85%	
	sell	-1,64%	
	sell	-1,72%	
	sell	-0,10%	
	sell	0,24%	
	sell	-0,24%	**volatility**
		-0,79%	**1,19%**

	2nd half 2007		
splitmin 35	*prediction*	*payoff*	
	hold	0,00%	
	hold	0,00%	
	sell	-1,26%	
	sell	-0,61%	
	sell	-0,72%	
	sell	-1,55%	
	sell	-0,64%	
	sell	0,95%	
	sell	0,96%	
	sell	1,79%	
	sell	2,02%	
	sell	-0,79%	
	sell	-0,96%	
	sell	-1,30%	
	sell	-2,63%	
	sell	-1,00%	
	sell	-1,25%	
	sell	-0,45%	
	sell	0,52%	
	sell	1,34%	
	sell	1,94%	
	sell	1,11%	
	sell	-0,70%	
	sell	-2,01%	
	sell	-2,97%	
	sell	-2,85%	
	sell	-1,64%	
	sell	-1,72%	
	sell	-0,10%	
	sell	0,24%	
	sell	-0,24%	**volatility**
		-14,51%	**1,35%**

	2nd half 2007		
splitmin 20	*prediction*	*payoff*	
	hold	0,00%	
	hold	0,00%	
	sell	-1,26%	
	sell	-0,61%	
	sell	-0,72%	
	sell	-1,55%	
	sell	-0,64%	
	sell	0,95%	
	sell	0,96%	
	sell	1,79%	
	sell	2,02%	
	sell	-0,79%	
	sell	-0,96%	
	sell	-1,30%	
	sell	-2,63%	
	sell	-1,00%	
	sell	-1,25%	
	sell	-0,45%	
	sell	0,52%	
	sell	1,34%	
	sell	1,94%	
	sell	1,11%	
	sell	-0,70%	
	sell	-2,01%	
	sell	-2,97%	
	sell	-2,85%	
	sell	-1,64%	
	sell	-1,72%	
	sell	-0,10%	
	sell	0,24%	
	sell	-0,24%	**volatility**
		-14,51%	**1,35%**

	2nd half 2007		
splitmin 50	*prediction*	*payoff*	
	hold	0,00%	
	hold	0,00%	
	sell	-1,26%	
	sell	-0,61%	
	sell	-0,72%	
	sell	-1,55%	
	sell	-0,64%	
	sell	0,95%	
	sell	0,96%	
	sell	1,79%	
	sell	2,02%	
	sell	-0,79%	
	sell	-0,96%	
	sell	-1,30%	
	sell	-2,63%	
	sell	-1,00%	
	sell	-1,25%	
	sell	-0,45%	
	sell	0,52%	
	sell	1,34%	
	sell	1,94%	
	sell	1,11%	
	sell	-0,70%	
	sell	-2,01%	
	sell	-2,97%	
	sell	-2,85%	
	sell	-1,64%	
	sell	-1,72%	
	sell	-0,10%	
	sell	0,24%	
	sell	-0,24%	**volatility**
		-14,51%	**1,35%**