SFB 649 Discussion Paper 2008-009

Recursive Portfolio Selection with Decision Trees

Anton Andriyashin*
Wolfgang Härdle*
Roman Timofeev*

* Humboldt-Universität zu Berlin, Germany

This research was supported by the Deutsche Forschungsgemeinschaft through the SFB 649 "Economic Risk".

http://sfb649.wiwi.hu-berlin.de
ISSN 1860-5664

SFB 649, Humboldt-Universität zu Berlin
Spandauer Straße 1, D-10178 Berlin

Recursive Portfolio Selection with Decision Trees*

Anton Andriyashin
CASE – Center for Applied Statistics and Economics
Humboldt-Universität zu Berlin,
Spandauer Straße 1, 10178 Berlin, Germany

Wolfgang K. Härdle
CASE – Center for Applied Statistics and Economics
Humboldt-Universität zu Berlin,
Spandauer Straße 1, 10178 Berlin, Germany

Roman Timofeev
CASE – Center for Applied Statistics and Economics
Humboldt-Universität zu Berlin,
Spandauer Straße 1, 10178 Berlin, Germany

January 15, 2008

*We gratefully acknowledge financial support by the Deutsche Forschungsgemeinschaft and the Sonderforschungsbereich 649 "Ökonomisches Risiko". Roman Timofeev's and Anton Andriyashin's research was supported by Deka Bank scholarship program.

Abstract

A great proportion of stock dynamics can be explained using publicly available information. The relationship between dynamics and public information may be of nonlinear character. In this paper we offer an approach to stock picking by employing so-called decision trees and applying them to XETRA DAX stocks. Using a set of fundamental and technical variables, stocks are classified into three groups according to the proposed position: long, short or neutral. More precisely, by assessing the current state of a company, which is represented by fundamental variables and current market situation, well reflected by technical variables, it is possible to suggest if the current market value of a company is underestimated, overestimated or the stock is fairly priced. The performance of the model over the observed period suggests that XETRA DAX stock returns can adequately be predicted by publicly available economic data. Another conclusion of this study is that the implied volatility variable, when included into the training sample, boosts the predictive power of the model significantly.

JEL classification: C14, C49, G11, G12

Keywords: CART, decision trees in finance, nonlinear decision rules, asset management, portfolio optimisation

1 Introduction

There is extensive literature in financial econometrics on the predictability of characteristics of stock prices (e.g. volatility, directions, duration) using publicly available information. Goyal & Welch (2003) investigated the empirical real-world out-of-sample performance of plain linear regressions to predict the equity premium and concluded that none of the combinations of the well-known and widely suggested variables (e.g. the dividend-price ratio, dividend yield, interest and inflation rates etc.) lead to an appropriate result. Unsatisfactory out-of-sample performance for linear regression was also obtained by other groups of researches such as Campbell (1988) and Fama & French (1988). Attempts to forecast the Eurex stock returns using Logit regression were undertaken by Amenc, et al (2003) – the model showed an average hit ratio of 2/3 over the observed period with the corresponding annual return of about 7%. Bauer & Molenaar (2002) employed a similar approach of Logit regression and provided information ratios greater than 0.50. Avramov (2002) used a Bayesian model averaging to develop a global model that has proven to be robust in predicting stock returns. As a result, a weighted model that averages across competing models provided the Sharpe ratios of less than 0.25.

In this paper an alternative approach to stock picking by employing Classification and Regression Trees (CART) is offered. Due to its properties (Breiman, et al 1987), CART provides considerable performance gains in comparison with other, more traditional models. CART is nonparametric, does not require variables to be selected in advance, is invariant to monotone transformations of the independent variables, is robust to the effects of outliers and, what most important is, provides superior performance to major stock exchange indexes. Stock pricing with CART has been considered among other methods like rolling regressions by the leading financial institutions, such as *SmithBarney* (a member of *Citigroup*), see Sorensen, et al (1999). The model yielded 19.62% (annualized) with the standard deviation of 11.96% and the Sharpe ratio of 1.23. *JPMorgan* (Seshadri 2003) used CART to classify US technology stocks into three classes: overpriced, underpriced and fairly priced ones. The performance of their quantitative model has shown 14.6% of the annualised return with the corresponding standard deviation of 9.5% and the Sharpe ratio of 1.54.

Our model based on weekly observations of XETRA DAX companies yielded 19.99% annu-

alised with the corresponding Sharpe ratio of 0.88, which clearly outperforms other benchmark strategies for the relevant period. Risk free rate was assessed by the three month London interbank offer rate (LIBOR).

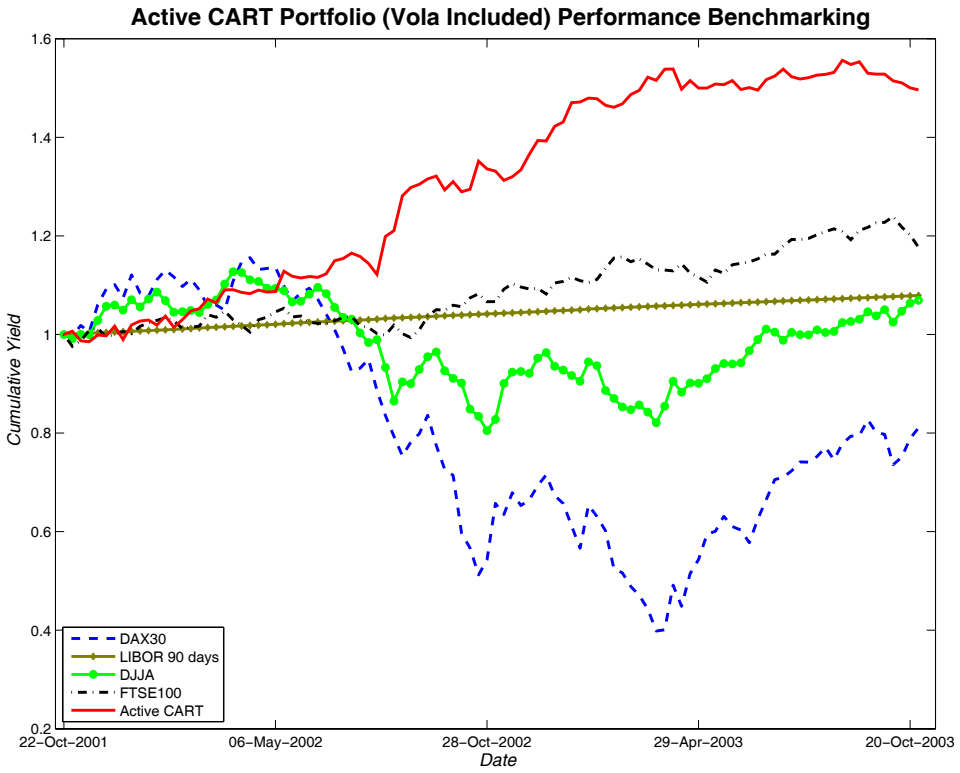

Figure 1.1: Wealth curves for active CART strategy and benchmark strategies

In the last part of the study we include the implied volatility variable in the learning sample that increased the performance of the strategy up to 25.55% per year with the corresponding Sharpe ratio of 1.59, see Figure 1.1. CART strategy (solid line) shows a superior performance in comparison with major stock exchange indexes and risk free rate. These results prove the importance of implied volatility as a risk factor that is consistent with the conclusions of Ferson & Harvey (1991) where it states that the relationship between stock and bond market returns and market fluctuations exists and is well approximated by the implied volatility.

The structure of the paper is as follows: Sections 2 and 3 give a short overview of the

decision tree methodology. We start with the general notions used in the study such as the splitting rule and the impurity measure, then in Section 3 the tree pruning mechanism is described. Section 4 describes available data and the construction of the learning sample, which is quite important in order to train adequately the decision trees. Section 5 provides the backtesting results and the model performance, Section 6 has concluding remarks.

2 CART – Classification and Regression Trees

CART stands for Classification and Regression Trees and is a nonparametric classification method that uses available data in the form of $(\mathcal{X}, \mathcal{Y})$. $\mathcal{X}$ is the matrix of explanatory variables and $\mathcal{Y}$ is the vector of classes which has to be defined *a priori*. It means that the available data may not contain the target characteristic $\mathcal{Y}$ in advance and has to be computed additionally – usually using the available data $\mathcal{X}$.

For stock picking applications of decision trees it is natural to regard an element of the class vector $y \in \mathcal{Y}$ as a predefined characteristic of a stock, for instance a stock could be subjectively *undervalued*, *overvalued* or it could also have a subjectively *fair price*. In this setup $y \in \{\text{long, short, neutral}\}$ are three predefined classes of stocks. $\mathcal{X}$ can then contain a set of fundamental and technical variables relating to a particular stock and probably some general macroeconomic factors as well. However, there could be several ways to assess the historical stock potential in the aforementioned terms and build vector $\mathcal{Y}$, this aspect will be examined in more detail in Section 4.

At this point we assume that each observation has its class that constitutes vector $\mathcal{Y}$ for the whole *learning sample* – the combination of available data from the past $\mathcal{X}$ and (probably computed additionally) target characteristic $\mathcal{Y}$. The data set $(\mathcal{X}, \mathcal{Y})$ is used to extract available data patterns – this is achieved by "learning" i.e. the creation of a decision tree T that extracts different outcomes from the past and tries to "explain" a connection between $\mathcal{X}$ and $\mathcal{Y}$ observed in the past in the form of a binary tree. The decision tree is then used to classify new data into classes from $\mathcal{Y}$ – in this way when new market information becomes available, for a given stock using the existing tree it is possible to produce a recommendation either to long, short or maintain the current position.

Consider an example of an artificially generated two-dimensional data set with five classes and the resulting decision tree presented in Figure 2.3. For the sake of simplicity, each color represents a class, and CART tends to separate the respective colour areas with the minimal number of questions (or splits) in a binary tree. Nodes with tags *Blue*, *Green*, *Black*, *Yellow* and *Purple* are the so called *terminal nodes* $\tilde{T}_k$ whereas the node at the top, with the question "Is $\mathcal{X}_1 \leq 0.5$?", is called the *root node*. If the answer is positive, the left branch of the tree is taken.

Decision trees are represented by a set of questions that splits the learning sample into smaller and smaller parts. CART asks only yes/no questions. A possible question could be: "Are the company's last reported Earnings Per Share (EPS) > 1.5?" or "Is the Brent crude oil 1-month future price < 80?". But where does the value of 1.5 in the question about EPS come from?

CART searches through all available variables and their possible values in order to find the *split* s – a combination of a variable from the available data $\mathcal{X}$ and the appropriate question value. The question s^* that splits the data into two parts with maximum homogeneity inside each of those parts is then selected as *optimal*. The process is repeated for each of the resulting data fragments since every question in a tree just splits the initial data set into two parts, see Section 2.2 for more details on splitting algorithm employed in this study. At some point of time a tree T reaches its "optimal" size, this means that no additional questions are added to the rule (refer to Section 3 for the description of the tree optimisation procedure). Finally, at the bottom of a tree there are terminal nodes $\tilde{T}_k$ that contain decision rule parts for a certain combination of data questions led to a particular node t. Different data questions lead to different terminal nodes, so a set of terminal nodes along with the respective paths to nodes constitute a final decision rule T^*.

The application of decision trees to a data set implies conducting three major steps:

- the construction of the so called *maximum tree* T_{MAX}
- the choice of the right tree size (tree pruning) T^*
- the classification of new data using the constructed tree T^*

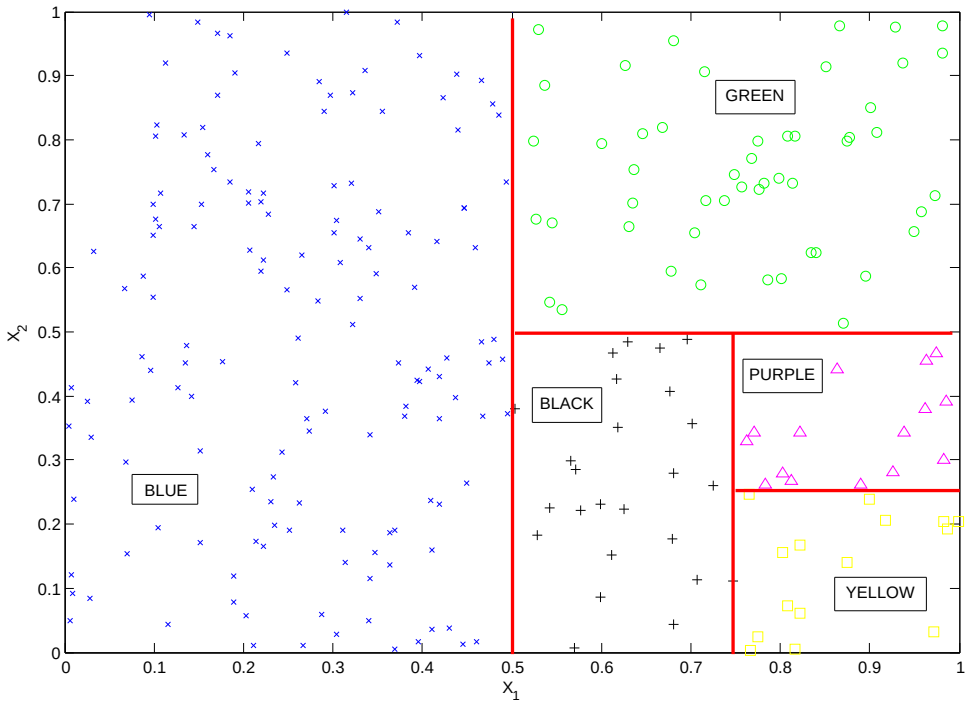

Figure 2.2: Application of CART to an artificial two-dimensional data set. Because of the special form of the questions (univariate linear splits), the separating lines are always orthogonal to the axes. For a given example four splits were sufficient to separate data into different nodes of the tree. The corresponding tree is depicted on Figure 2.3

2.1 Construction of the Maximum Tree

Let P be the number of variables from the available company data $\mathcal{X}$. If $\mathcal{X} = (\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_P)$ is the matrix of the learning sample and there are M observations available, then the class vector $\mathcal{Y}$ has the same length M, i.e. a class tag y_i is assigned to each i-th observation of the learning sample. Without the loss of generality let us also suppose that there are J unique classes (the situation when $y_i \in \{\text{long, short, neutral}\}$ corresponds to $J = 3$).

Let t_P be a parent node and t_L, t_R – left and right child nodes of the parent node t_P respectively so that a fraction p_L of observations from node t_P follows to the left child node and a fraction $p_R = (1 - p_L)$ – to the right one. If n_P is the number of observations in t_P

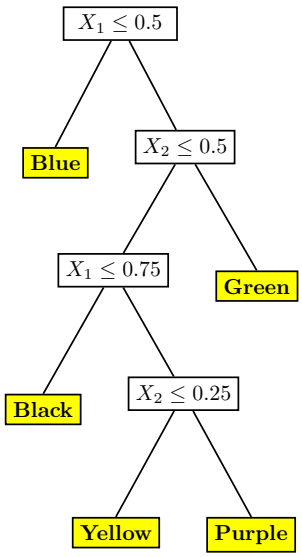

Figure 2.3: The resulting classification tree. Left branches stand for the positive answers, right branches – for the negative ones. There are four splits and five terminal nodes in this tree.

and n_L, n_R – in t_L and t_R respectively, then

$$p_L = \frac{n_L}{n_P}, \quad p_R = \frac{n_R}{n_P} \tag{1}$$

A classification tree is built in accordance with a *splitting rule* – a rule that determines a split s^* at each node. Its aim is to create two more homogenous groups by splitting the initial less homogenous one (the parent node) into two parts (two child nodes). A split s^* contains those variable $\mathcal{X}_{p^*}, p^* \in \{1, \ldots, P\}$ from the matrix of explanatory variables $\mathcal{X}$ and a question value x^*, which lead to splitting of the parent node t_P into nodes t_L and t_R, when a question "Is $X_{p^*} < x^*$?" separates the data contained in t_P into two different groups with the maximum feasible inner homogeneity.

Homogeneity is defined via an *impurity function* $i(t)$. This is an arbitrary function that has a unique maximum at point $\left(\frac{1}{J}, \frac{1}{J}, \ldots, \frac{1}{J}\right) \in \mathbb{R}^P$ and a unique minimum at points $(1, 0, 0, \ldots, 0)$, $(0, 1, 0, \ldots, 0)$, …, $(0, 0, 0, \ldots, 1) \in \mathbb{R}^P$. It also possesses some important

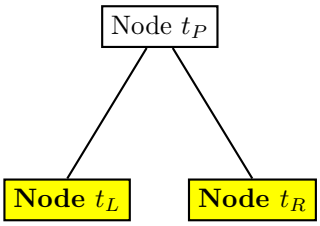

Figure 2.4: Parent and child node hierarchy

technical properties, refer to Breiman, et al (1987) for more details. Section 2.2 contains the explicit definition of this function that is frequently employed for applied decision tree analysis and was employed in this study.

The split corresponding to the maximum homogeneity of the left and right child nodes compared to the parent node is equivalent to the split following from the maximization of change of the impurity function $\Delta i(t)$ for an arbitrary node t and arbitrary split s:

$$\Delta i(t) = i(t_P) - \mathbb{E}\left\{i(t_C)\right\} \tag{2}$$

where $\mathbb{E}\left\{i(t_C)\right\} = p_L i(t_L) + p_R i(t_R)$ and set of child nodes $C = \{L, R\}$.

Assuming that p_L and p_R are the estimated probabilities of the right and left nodes (respective proportions of observations distributing from the parent node to two child nodes), it follows that:

$$\Delta i(s,t) = i(t) - p_L i(t_L) - p_R i(t_R) \tag{3}$$

for an arbitrary data split s.

Therefore, at each node one solves the following optimisation problem:

$$\begin{aligned} s^* = \underset{s}{\operatorname{argmax}}\, \Delta i(s,t) = \underset{s}{\operatorname{argmax}}\left\{-p_L i(t_L) - p_R i(t_R)\right\} = \\ = \underset{s}{\operatorname{argmin}}\left\{p_L i(t_L) + p_R i(t_R)\right\} \end{aligned} \tag{4}$$

Note that t_L and t_R are implicit functions of s since a change of an arbitrary question variable $\mathcal{X}_{\tilde{p}}, \tilde{p} \in \{1, \ldots, P\}$ or an arbitrary question value $\tilde{x}$ (change of an arbitrary $\tilde{s}$: "Is $\mathcal{X}_{\tilde{p}} < \tilde{x}$?") makes the values t_L and t_R change as well.

In this algorithm CART searches through all possible values of all variables constituting the matrix $\mathcal{X}$ for the best split s^* that maximises the change of impurity function $\Delta i(s^*, t)$.

The *maximum tree* T_{MAX} is the tree containing the maximum number of nodes for a given data set. Put differently, it is a tree built by applying equation (4) to the original data set and resulting split data portions until the following condition holds. This condition defines T_{MAX} as the tree where each terminal node contains only observations belonging to the same class j:

$$\forall t \in \tilde{T} \;\; \exists j \colon p(j|t) = 1 \tag{5}$$

where $\tilde{T}$ is the set of terminal nodes of a tree T.

The next important step is to define the impurity function $i(t)$. Although impurity functions can be defined in numerous ways, the Gini index is the preferred choice in financial applications, see Kolyshkina & Brookes (2002).

2.2 Gini Splitting Rule

Employing the idea of the Gini index, this special form of the impurity function $i(t)$ can be written down as follows:

$$i(t) = \sum_{\substack{k \neq l \\ k=1}}^{J} \sum_{\substack{l \neq k \\ l=1}}^{J} p(k|t)p(l|t) \tag{6}$$

where $k, l = \overline{1, J}$ are class indices and

$$p(j|t) = \frac{n_t(j)}{n_t} \tag{7}$$

where $n_t(j)$ is the number of observations from $\mathcal{X}$ belonging to the class $j \in \{1, \ldots, J\}$ that have been filtered to a node t for a given split s; n_t is the overall number of observations contained in the node t.

Applying the Gini splitting rule to (3), $\Delta i(t)$ transforms to:

$$\Delta i(t) = -\sum_{j=1}^{J} p^2(j|t) + p_L \sum_{j=1}^{J} p^2(j|t_L) + p_R \sum_{j=1}^{J} p^2(j|t_R) \tag{8}$$

and (4) takes the following form:

$$s^* = \underset{s}{\operatorname{argmax}} \left\{ -\sum_{j=1}^{J} p^2(j|t) + p_L \sum_{j=1}^{J} p^2(j|t_L) + p_R \sum_{j=1}^{J} p^2(j|t_R) \right\} \tag{9}$$

It is possible to show that for a certain setup the Gini index is equivalent to the data variance according to the class in $\mathcal{Y}$ i.e. it evaluates class homogeneity of the data in a given node, see Breiman, et al (1987) for more details. The Gini algorithm usually tends to search for the largest class in a learning sample and isolate it from the rest of the data – the relevant examples and the comparison with other impurity function types can also be found in Breiman, et al (1987).

3 Optimal Tree Size

3.1 Over- and Underparameterization of the Trees

The algorithm of the tree building is as follows. Equation (9) is first applied to the whole learning sample (root node) $\mathcal{X}$, after which it is applied to each of the created tree nodes. This process can be looped until either $i(t) = 0$ for every terminal node as indicated in (5) or until the size of the tree becomes balanced.

But what is the balanced or optimal size of a tree? And why is a maximum tree not always the best choice?

Indeed, applying (9) until (5) holds means that at each step it was possible to decrease the class heterogeneity inside the learning sample by filtering out observations of other classes and assigning them to other nodes. Therefore, if in the limiting case each observation can be assigned to a separate node, it would only mean that even smallest random disturbances including, but not limited to measurement errors, were represented as parts of a final decision rule. Clearly one would wish to have only those reliable parts of the tree that stand for a fundamental inner data pattern. And the reason is straightforward – new data to classify would have to pass through the created tree, therefore if they pass through a noisy part of the rule, with a high probability the classification of new data may be wrong.

On the other hand, a small tree is also not a panacea since it can be under-parameterised, i.e. it does not account for significant data portions in the learning sample.

One solution to the problem could be the application of the *cross-validation* to the subtrees of different sizes and comparison of their performance.

3.2 Cost-complexity Function and Cross-validation

The idea of this method presented in Breiman, et al (1987) is to introduce some new measure that would be able to take into account *tree complexity*, i.e. its size which can be estimated by the number of terminal nodes. Then the maximum tree is penalized for its big size, however on the other hand it makes perfect in-sample predictions. Small trees, of course, get a much lower penalty for their size, but their predicting abilities are naturally limited. The aim is therefore to find a balance between the tree size, which is penalized, and the predictive power of the tree. This can be achieved via the cost-complexity function to be defined later in this Section.

First, let us define the *internal misclassification error* of an arbitrary observation at node t as $e(t) = 1 - \max_j p(j|\, t)$, define also $E(t) = e(t)p(t)$. Then the *internal misclassification tree error* is:

$$E(T) = \sum_{t \in \tilde{T}} E(t) \tag{10}$$

where $\tilde{T}$ is a set of terminal nodes. For any subtree $T \leq T_{MAX}$ define the number of terminal nodes $\left|\tilde{T}\right|$ as the measure of its complexity. Then the cost-complexity function aimed to optimize the decision tree size is defined as follows:

$$E_\alpha(T) = E(T) + \alpha \left|\tilde{T}\right| \tag{11}$$

where $\alpha \geq 0$ is the complexity parameter and $\alpha \left|\tilde{T}\right|$ is the cost component.

Although α can have an infinite number of values, the number of subtrees of T_{MAX} resulting in minimisation of $E_\alpha(T)$ is *finite*. Hence pruning of T_{MAX} leads to creation of subtrees sequence $T_1, T_2, T_3, \ldots$ with a decreasing number of terminal nodes. Since the sequence is finite, if $T(\alpha)$ is an optimal subtree for some arbitrary α, then it will remain optimal until the complexity parameter is not changed to some α' when $T(\alpha')$ becomes a new optimal subtree until complexity parameter value is α'' and so on.

In Breiman, et al (1987) it is shown that for $\forall \alpha \geq 0$ an optimal tree $T(\alpha)$ exists in the sense that

1. $E_\alpha \{T(\alpha)\} = \min\limits_{T \leq T_{MAX}} E_\alpha(T) = \min\limits_{T \leq T_{MAX}} \left[E(T) + \alpha \left|\tilde{T}\right|\right]$
2. if $E_\alpha(T) = E_\alpha \{T(\alpha)\}$, then $T(\alpha) \leq T$.

Let $\{t_0\}$ denote the root node. This way one can get a sequence of optimal *nested* subtrees $T_{MAX} \succ T_1 \succ T_2 \succ T_3 \succ \ldots \succ \{t_0\}$ for which it is possible to prove that the sequence $\{\alpha_k\}$ is increasing, i.e. $\alpha_k < \alpha_{k+1}$, $k \geq 1$ and $\alpha_1 = 0$. For $k \geq 1$: $\alpha_k \leq \alpha < \alpha_{k+1}$ and $T(\alpha) = T(\alpha_k) = T_k$.

Applying then the method of the V-fold cross-validation to the sequence $T_{MAX} \succ T_1 \succ T_2 \succ T_3 \succ \ldots \succ \{t_0\}$, an *optimal tree* is then determined.

However, selecting an optimal tree as the one with the minimum value of $E^{CV}(T)$ may not be the best solution since usually there is a whole range of values $E^{CV}(T)$ satisfying $E^{CV}(T) < E^{CV}_{MIN}(T) + \varepsilon$ for small $\varepsilon > 0$. And if V is less than the number of observations in $\mathcal{X}$, then the second run of V-fold cross-validation procedure could provide slightly different results because of the randomness embedded in the algorithm of cross-validation.

Therefore, a so called *one standard error* empirical rule is applied. It states that if T_{k_0} is the tree minimizing $E^{CV}(T_{k_0})$ from the sequence of nested subtrees $T_{MAX} \succ T_1 \succ T_2 \succ T_3 \succ \ldots \succ \{t_0\}$, then a value k_1 and a correspondent tree T_{k_1} are selected so that

$$\underset{k_1}{\operatorname{argmax}}\, \hat{E}(T_{k_1}) \leq \hat{E}(T_{k_0}) + \sigma\left\{\hat{E}(T_{k_0})\right\} \tag{12}$$

where $\sigma(\cdot)$ denotes the sample estimate of the standard error and $\hat{E}(\cdot)$ – the relevant estimates of the internal misclassification errors (introduced in (10)) that are derived from data subsamples employed by the cross-validation procedure.

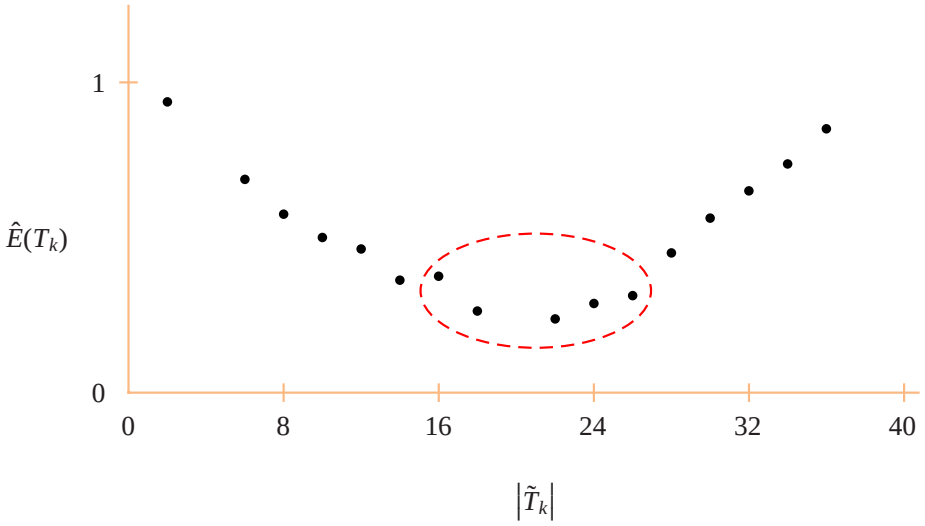

Figure 3.5: The example of relationship between $\hat{E}(T_k)$ and number of terminal nodes

The dotted line in Figure 3.5 shows the area where the values of $\hat{E}(T_k)$ only slightly differ from $\min_{|\tilde{T}_k|} \hat{E}(T_k)$. The left edge, which is roughly equivalent to 16 terminal nodes, shows the application of the one standard error rule. The use of the one standard error rule allows not only more robust results to be achieved, but also to get trees of lower complexity given the error comparable with $\min_{|\tilde{T}_k|} \hat{E}(T_k)$.

4 Available Data and Calibration

In this study we operate with a single data set of XETRA DAX companies for the period of 27 April, 2000 – 30 October, 2003 that, along with historical stock prices, has a set of technical and fundamental indicators. Table 4.1 provides an overview of the available data.

Indicator	Type	Frequency	Description
Momentum	Technical	1 day	$M_t = P_t - P_{t-T},\ T = 20$
Stochastic	Technical	1 day	$\frac{P_t - P_L}{P_H - P_L},\ P_H = max(P_t),\ P_L = min(P_t)$
MA	Technical	1 day	$MA(T) = \frac{\sum_{i=t-T}^{t} P_i}{T},\ T = 12$
MA St. Error	Technical	1 day	Standard deviation of MA
MACD	Technical	1 day	$(1 - \frac{n_1}{n_2})\{MA(n_1) - MA(n_2 - n_1)\}$ $n_1 = 12,\ n_2 = 26$
ROC	Technical	1 day	$\frac{P_t}{P_{t-T}},\ T = 10$
TRIX	Technical	1 day	Triple exponentially smoothed MA
BV	Fundamental	1 month	Book Value
CF	Fundamental	1 month	Cash Flow
Dividends paid	Fundamental	1 month	-
			Depreciation
EPS	Fundamental	1 month	Earnings Per Share
Sales	Fundamental	1 month	-
ImplVola	Fundamental	1 day	Implied volatility

Table 4.1: List of available variables, t is the current time period

The data contain two types of variables: technical indicators, which are collected daily and usually represent some derivative of historical price, and fundamental data, which provide us with the evidence of the current market status of the company by means of various numbers and ratios from the balance sheet and the income statement. Fundamental data are collected monthly and sometimes quarterly. In order to have the same time scale, the following transformations were applied: other than daily variables are normalized by the stock price, e.g. instead of BV_t, $\frac{BV_t}{P_t}$ is taken, where t refers to a particular time moment, see Table 4.1 for the description of the available variables.

Because of the different measurement periods for technical and fundamental data (refer to Table 4.1), it is an open question which time scale to use for the stock picking analysis. Obviously, daily data provide us with a sufficient number of observations in a learning set, but at the same time they may contain undesired market fluctuations and natural noise (e.g. measurement errors). Since fundamental data are collected monthly, their daily changes bring us little new information. On the other hand, monthly data will give us 12 observations per year, which is not enough to construct a reliable tree. Our analysis is therefore based on *weekly observations* which is a trade-off between data scarcity and informational content. The final data set consists of technical data by the end of each week and fundamental indicators normalised by corresponding stock prices.

Individual trees are built for each stock from XETRA DAX. This is opposed to a setup when the data from all the stocks are combined and a single common tree is employed to provide forecasts for each of the stocks. However, due to the different nature, scale and business areas of the companies in XETRA DAX, individual decision trees are employed to capture those company peculiarities that might be lost if a single common tree is used.

The available data are divided into three groups: the learning set, the test/calibration set and the validation set. The first data set is a learning sample that is used to construct the initial decision tree for every company. The test set is employed for the model parameters to be optimised. And finally, with the optimised model parameters, the simulated performance of each stock is evaluated using the last section of the data – the validation set.

In Section 2 the stock classes (stored in the vector $\mathcal{Y}$) were introduced in the form of three possible values – *long*, *short* or *neutral* standing for undervalued, overvalued and fairly priced stocks. In this study we assess the relative performance of a stock in the following way.

Let $\bar{R}$ be some positive threshold. Then for given $\bar{R}$ the tree classes are defined as following:

$$\begin{cases} R_t \geq \bar{R} & \Rightarrow \textit{long} \\ -\bar{R} \leq R_t \leq \bar{R} & \Rightarrow \textit{neutral} \\ R_t \leq -\bar{R} & \Rightarrow \textit{short} \end{cases}$$

If the forward-looking one period stock return for the next period $t+1$ computed at time t

$$R_t = \frac{P_{t+1} - P_t}{P_t} \tag{13}$$

exceeds the threshold value $\bar{R}$ (where P_t is the current stock price), this stock is regarded as underperforming in the *current period* t.

The value of $\bar{R}$ is supposed to be different for each of the analysed XETRA DAX stocks. Setting it up too high $\bar{R}$ results in a low number of cases in the learning sample where active positions should be maintained – the trading system is too cautious. On the other hand, if $\bar{R}$ is close to zero, then even small price fluctuations, which are, perhaps, due to only speculative market activity and not fundamental reasons, trigger the signal for maintaining the active position – in this case the trading system is too sensitive to market signals.

The threshold value $\bar{R}$ was optimised using the calibration set for each stock individually. It was chosen in such a way that maximises the return of the stock over the calibration period if the trading activity is simulated. For that purpose different values of $\bar{R}$ starting from 0% and ending at 3% with a step of 0.25% were analysed resulting in 13 different scenarios for each stock. Although different stocks resulted in various values of $\bar{R}$, in the majority of cases $\bar{R}$ was close to 1.5%.

The size of the learning set was set to one year (52 weekly observations). In order to reflect structural changes on the market, a sliding window approach was employed, i.e. as new data point becomes available, it was included in the sample while the oldest one was excluded. Thus, the size of the learning sample remains fixed over time, but it is constantly updated with the new observations as they become available.

The calibration sample size was set to 26 weeks. The rest data points were allocated for the validation set.

5 Backtesting Results

As mentioned before, each available stock from XETRA DAX was analysed using the individual trees. Due to the data scarcity only the stocks with the market data available during

the whole period were regarded: ADS (1939), VOW (621), SAP (24243), DTE (19594), BMW (708), FME (684), HEN3 (3917), SIE (657) – the numbers in parentheses correspond to the relevant company XETRA DAX codes.

For the validation period for every time point the model forecasted position (via optimised and properly calibrated decision trees) was compared with an actual following period return happened in the past. If the position coincided with the forward-looking stock price change direction, the trading system recorded the relevant profit.

An example of the decision tree for SIE (657) is given in Figure 5.6. Left branch corresponds to the positive answer, whereas right branch to the negative answer to the question in the parent node. The number of observations for each of three classes is given in brackets. The class of terminal nodes (marked with yellow) is defined by the dominating class of the node. Since the sliding window approach was employed, a new tree was constructed as soon as data become available.

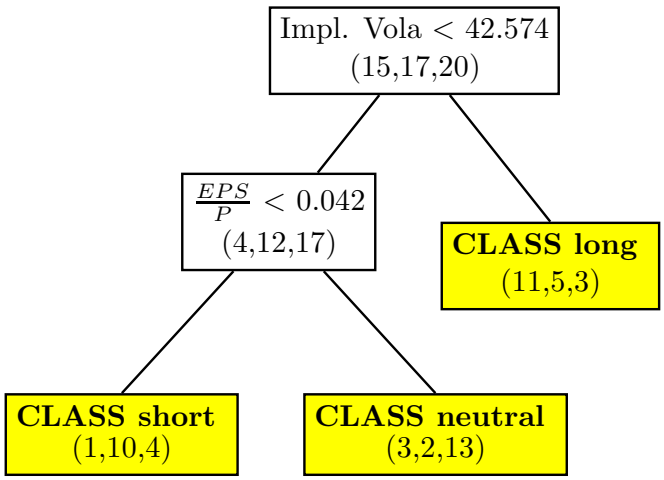

Figure 5.6: One of the decision trees for SIE (657), $\bar{R} = 3\%$

At the beginning of each period all open positions were closed and profits were not reinvested. Transaction costs in amount of 10 b.p. for each operation were included into the calculations.

An equally-weighted portfolio out of the available stocks with the recommended active positions (either long or short) was formed for every period. Because the trading system updated

the recommendations every week (when new market data became available), the weights of the portfolio were recalculated as well. If $A_t > 0$ is the number of stocks with the recommended active position for the current period t, then the relevant i-th stock weight in the portfolio at time point t is ω_{it}:

$$|\omega_{it}| = \frac{1}{A_t}. \tag{14}$$

If $A_t = 0$, then the portfolio positions remain unchanged from the previous period. The portfolio is created at the first week c of the validating period when $A_c > 0$.

The creation of the equally weighted portfolio is a common way to get the backtesting results for stock picking in the case of multiple stocks, see, for example, Amenc, et al (2003), Seshadri (2003) or Sorensen, et al (1999).

Figure 5.7 shows the weekly portfolio returns and Figure 5.8 plots the wealth curves of the CART strategy and traditional benchmarks: dynamics of XETRA DAX, FTSE 100 and Dow Jones Industrial indices. The risk free rate was approximated with the three month LIBOR interest rate. For the observed period the average annualized profit was 19.99% with the corresponding Sharpe ratio of 0.88.

	Value
Sharpe ratio	0.88
Mean relative weekly	19.99%
Skewness	0.63
Kurtosis	6.69
Risk free (Avg. LIBOR)	0.04

Table 5.2: Performance statistics for portfolio without the implied volatility variable in the learning sample

To test the relevance of the implied volatility in terms of the stock price predicting power, the second recursive portfolio based on the learning samples with the implied volatility variable was constructed. Weekly returns and wealth curve are available in Figures 5.9 and 5.10. The performance of the portfolio with the implied volatility variable is summarised in Table 5.3. Because the implied volatility data were not available for SAP (24243), we excluded this

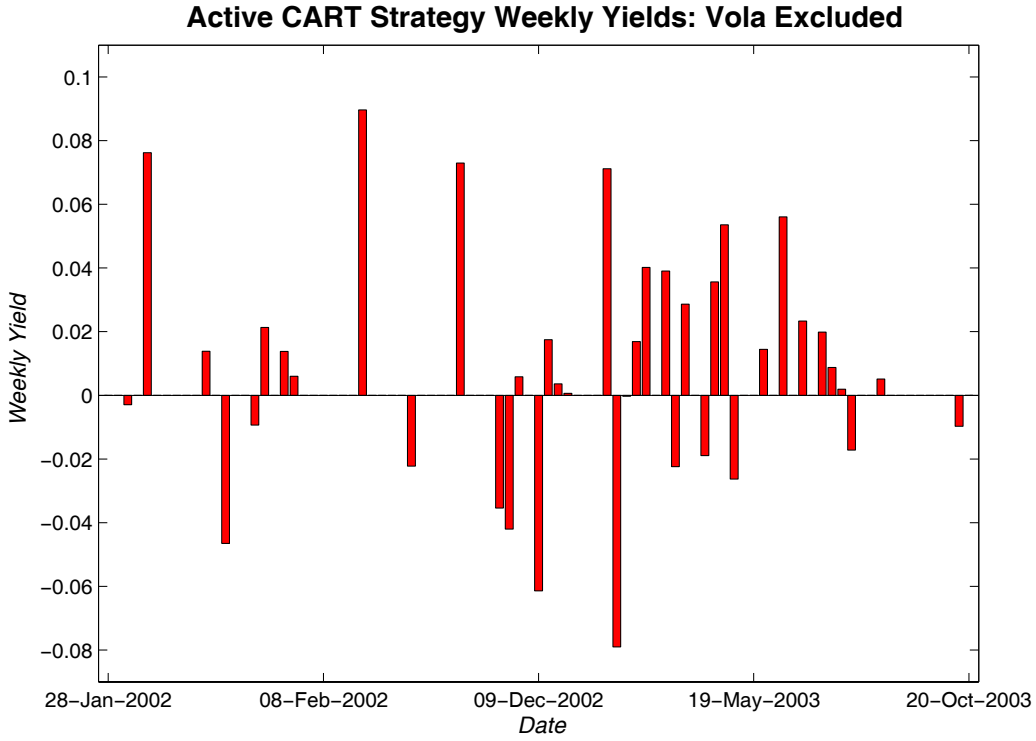

Figure 5.7: Weekly portfolio returns for active CART strategy without the implied volatility variable in the learning sample

stock from the second portfolio. Hence one can notice slightly different backtesting periods on the following pairs of Figures: 5.7, 5.8 vs 5.9, 5.10.

	Value
Sharpe ratio	1.59
Mean relative weekly	25.55%
Skewness	1.00
Kurtosis	5.68
Risk free (Avg. LIBOR)	0.04

Table 5.3: Performance statistics for portfolio with the implied volatility in the learning sample

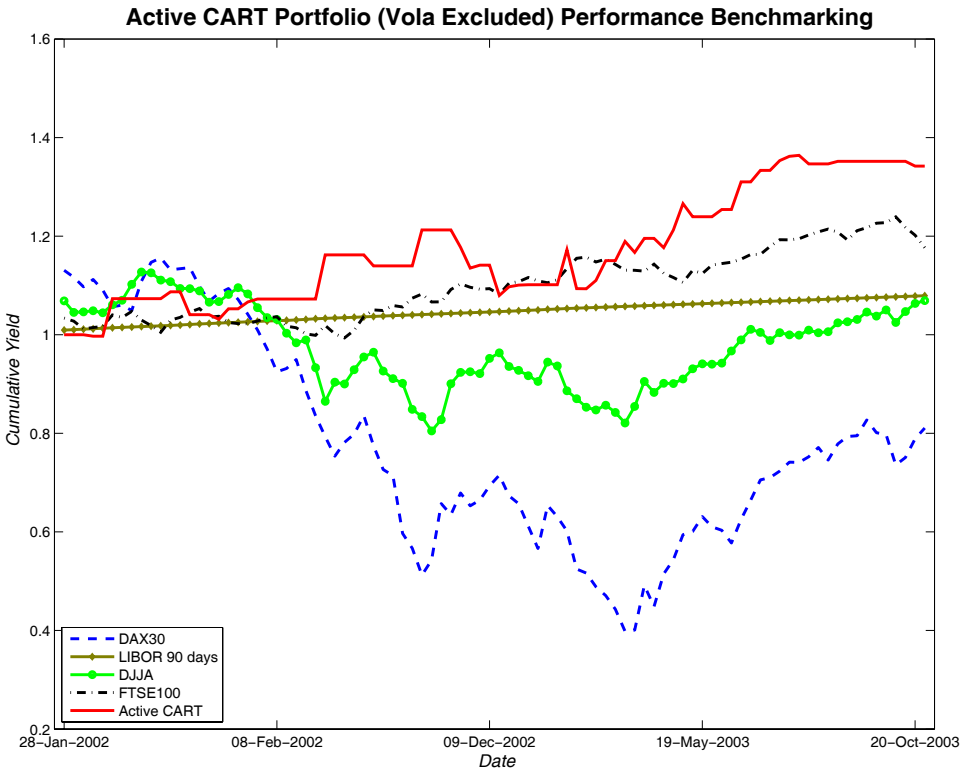

Figure 5.8: Wealth curves for active CART strategy without the implied volatility variable in the learning sample; wealth curves for benchmark strategies

It is clear that the implied volatility variable, at least for the indicated stock market and period, had a positive effect on the stock price predictive power. Being one of the factors that allows the measurement of the overall market uncertainty and, ultimately, one of the important risk factors, the inclusion of implied volatility variable in the learning sample boosted the forecasting potential of the proposed trading model based on binary decision trees.

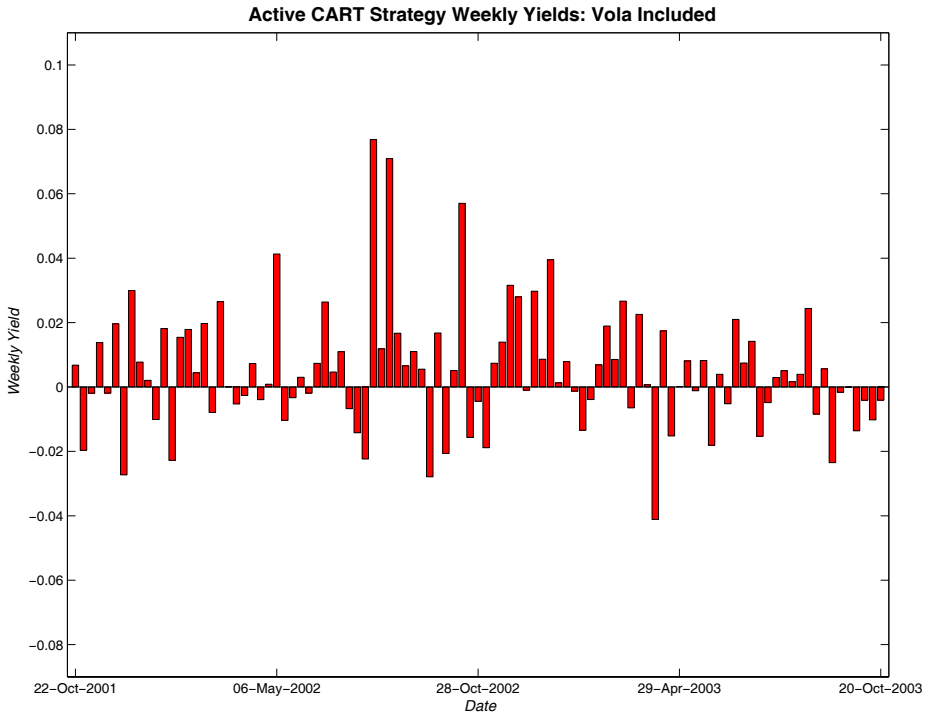

Figure 5.9: Weekly portfolio returns for active CART strategy with the volatility variable in the learning sample

6 Concluding Remarks

In this study it is assumed that stock selection can effectively be based on the proper analysis of available market data, i.e. a relationship of an unknown non-linear form between the current stock prices and the lagged market indicators exists. This relationship is estimated here via a nonparametric classification method called decision trees. Decision trees are a type of the classification rule that describes the relationship between the stock price and available market information in the form of a binary tree and are further employed to predict the future movements of the price. We also imply that this dependency may change in time with the evolving financial markets and therefore rebuild the decision tree for each period (week) whereas the last observations are included into the learning sample to keep it updated

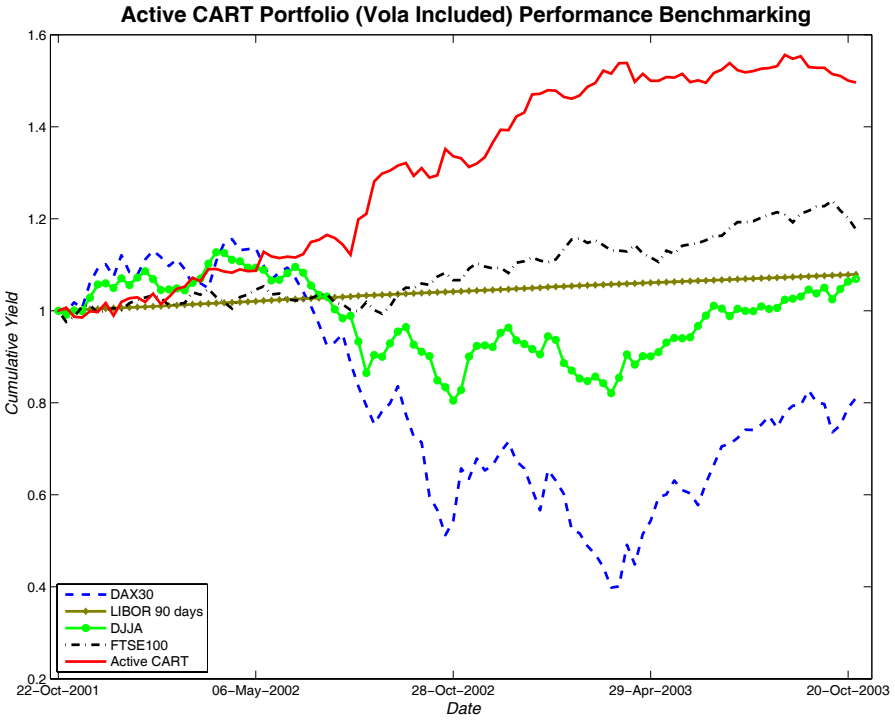

Figure 5.10: Wealth curves for active CART strategy with the implied volatility variable in the learning sample; wealth curves for benchmark strategies

and the oldest one is excluded to maintain the fixed sample size.

In order to evaluate the importance of the market risk factor for the stock prediction power, two different recursive portfolios – including and excluding the implied volatility variable in the learning samples – were created. The first portfolio was constructed using only technical and fundamental information available from data providers. With an annualised yield of 19.99% and the corresponding Sharpe ratio of 0.88 it clearly outperformed traditional benchmark strategies. Having included the implied volatility variable, the backtesting performance of the trading system was boosted up to an annual yield of 25.55% and the Sharpe ratio of 1.59. This result is consistent with previous research in the area implying that equity premium can be well explained by risk factors that are well approximated by the implied volatility.

References

Amenc, N., Malaise P., Martellini, L. & Sfeir, D. (2003). Portable Alpha and Portable Beta Strategies in the Eurozone, *Risk and asset management research center, EDHEC business school*.

Avramov D. & Chordia F. (2006). Predicting stock returns, *Journal of Financial Economics 82*: 387-415.

Avramov D. (2002). Stock return predictability and model uncertainty *Journal of Financial Economics 64, Nm. 3*: 423-458.

Bauer, R. & Molenaar, R. (2002). Is the Value Premium Predictable in Real Time? *Working paper, Maastricht University*

Breiman, L., Friedman, H. J., Olshen, A. R., Stone, J. C. (1987). Classification and regression trees, *The Wadsworth Statistics/Probability Seriesy*: 358 p.

Campbell, J. Y. & Shiller, R. J. (1988). Stock Prices, Earnings, and Expected Dividends, *Journal of Finance 43 Nm.3*: 661-676.

Campbell, J. Y. (1987). Stock returns and the Term Structure, *Journal of Financial Economics 18 Nm. 2*: 373-399.

Goyal, A. & Welch, I. (2003). Predicting the Equity Premium with Dividend Ratios, *Management Science 49 Nm. 5*: 639-654.

Fama, E. F. & French, K. R. (1988). Dividend Yields and Expected Stock Returns, *Journal of Financial Economics 22 Nm. 1*: 3-25.

Fama, E. F. & French, K. R. (1988). Permanent and Temporary Components of Stock Prices, *Journal of Political Economy 96 Nm. 2*: 246-73.

Ferson, W. & Harvey, C. (1991). The variation in economic risk premia, *Journal of Political Economy 99*: 385-415.

Hafner, M. C. & Herwatz, H. (1999). Time-Varying Price of Risk in the CAPM-Approaches, Empricial Evidence and Implications, *JFinance 19 Nm. 2*: 93-112.

Inna Kolyshkina & Richard Brookes. (2002). Data mining approaches to modelling insurance risk, *electronic version, PricewaterhouseCoopers*, 22 October 2002, 20 p.

Seshadri, L. (2003). JPMorgan US Quantitative Factor Model: August 2003 Stock Lists, *electronic version, JPMorgan Quantitative Equity and Derivatives Strategy*: 7 p.

Sorensen, H. E., Ooi, K. C. & Miller, L. K. (1999). The Decision Tree Approach to Stock Selection, *Salomon Smith Barney Equity Research: United States, Global Quantitative Research*: 28 p.

SFB 649 Discussion Paper Series 2008

For a complete list of Discussion Papers published by the SFB 649, please visit http://sfb649.wiwi.hu-berlin.de.

001 "Testing Monotonicity of Pricing Kernels" by Yuri Golubev, Wolfgang Härdle and Roman Timonfeev, January 2008.
002 "Adaptive pointwise estimation in time-inhomogeneous time-series models" by Pavel Cizek, Wolfgang Härdle and Vladimir Spokoiny, January 2008.
003 "The Bayesian Additive Classification Tree Applied to Credit Risk Modelling" by Junni L. Zhang and Wolfgang Härdle, January 2008.
004 "Independent Component Analysis Via Copula Techniques" by Ray-Bing Chen, Meihui Guo, Wolfgang Härdle and Shih-Feng Huang, January 2008.
005 "The Default Risk of Firms Examined with Smooth Support Vector Machines" by Wolfgang Härdle, Yuh-Jye Lee, Dorothea Schäfer and Yi-Ren Yeh, January 2008.
006 "Value-at-Risk and Expected Shortfall when there is long range dependence" by Wolfgang Härdle and Julius Mungo, Januray 2008.
007 "A Consistent Nonparametric Test for Causality in Quantile" by Kiho Jeong and Wolfgang Härdle, January 2008.
008 "Do Legal Standards Affect Ethical Concerns of Consumers?" by Dirk Engelmann and Dorothea Kübler, January 2008.
009 "Recursive Portfolio Selection with Decision Trees" by Anton Andriyashin, Wolfgang Härdle and Roman Timofeev, January 2008.

SFB 649, Spandauer Straße 1, D-10178 Berlin
http://sfb649.wiwi.hu-berlin.de

This research was supported by the Deutsche Forschungsgemeinschaft through the SFB 649 "Economic Risk".