Erasmus University Rotterdam
Erasmus School of Economics
Bachelor Thesis Bsc^2 in Econometrics and Economics

Stock Market Forecasting using Machine Learning: a Heteroscedastic Perspective

P.K. Pel (525573)

Supervisor:	A. Venes Schmidt
Second assessor:	O. Kleen
Date final version:	July 2nd 2023

1 Introduction

Accurately predicting the movement of stock prices has interested investors ever since the first stock market opened. Nowadays, technical analysis, the prediction of prices using past market data, is used widely by professionals and amateurs alike to seek larger profits than can be achieved by following the market (Menkhoff, 2010). However, the idea that it is possible to accurately predict and thus systematically outperform the market without taking on more risk is in direct contradiction with the efficient market hypothesis first proposed in Fama (1970). Therefore, it is not yet clear whether technical analysis can actually yield excess returns under the same risk.

This paper compares the forecasting performance of several classical and machine learning models when applied to a technical analysis setting. As input the models receive the (lagged) intraday open, close, high and low prices of the stock alongside the volume and the time of the day, which are used to predict the change in the closing price of the largest stocks in the S&P 500 index. Also, using these forecasts the performance of a basic trading algorithm is tested to show if it is possible to beat the market using technical analysis. Additionally, data on used car prices, alcohol use and fishery is analysed to show the potential performance differential of the different models in a less noisy setting.

As a classical benchmark model a random walk (RW) is considered, while the machine learning models that are used are random forest (RF), eXtreme Gradient Boosting (XGBoost), Bayesian additive regression trees (BART) and heteroscedastic BART (HBART). These machine learning models are all tree-based, which are used with increasing frequency in economic forecasting (Masini et al., 2023). The stock market is an area where nonlinearities can play a prominent role (Caginalp & DeSantis, 2011). The tree-based methods can potentially use these nonlinearities to provide better predictions than classical models. BART is considered in particular, since the Bayesian trees allow for better tuning of hyperparameters than RF and XGBoost, and is able to make predictions outside of the range of data, unlike RF and XGBoost. HBART extends BART by modelling the variance alongside the mean as variable-dependent whereas BART only models a constant variance. Hence, HBART introduces heteroscedasticity to the model. The stock market is known to have a time-dependent variance (Nishiyama, 1998). Therefore, HBART may be able to use both the nonlinearities and the heteroscedasticity in the data to give better forecasts compared to the homoscedastic BART.

Predicting the stock market is widely regarded as a difficult task due to the high level of noise in financial time series and the generally accepted efficient market hypothesis (Fama, 1970). Nevertheless, throughout the past decades numerous attempts have been made to outperform the random walk in stock market prediction, which is unbeatable under the efficient market hypothesis (Fama, 1995). Since 2015, the use of AI and machine learning in the financial literature has increased substantially (Ahmed et al., 2022). In the last few years, the use of machine learning in the financial sector has also surged (Lakhchini et al., 2022). The stock market has also seen applications of machine learning in recent times. XGBoost and RF have been used successfully in forecasting the direction of stock prices using a selection of technical indicators for a time horizon between 3 and 90 days (Basak et al., 2019). Using high-frequency data, a deep learning method was able to effectively predict prices a few minutes ahead (Lanbouri

& Achchab, 2020). Machine learning methods, including RF, have been used to improve the performance of technical analysis too (Ayala et al., 2021). In macroeconomic forecasting, RF has been used with great success like in Medeiros et al. (2021), where it robustly outperformed a great quantity of machine learning models in forecasting US inflation. BART has been used with success in the economic literature for causal inference problems (Athey & Imbens, 2019).

This paper contributes to the existing literature concerning machine learning in finance in several ways. Firstly, it can be tested whether machine learning methods can outperform a random walk in predicting the hourly change in the closing price of a stock. Secondly, it can be tested whether by introducing heteroscedasticity to the BART model, HBART is able to give better predictions for stock market data. Thirdly, by testing the out-of-sample stock market predictions using the trading algorithm, their effectiveness can be tested. Fourthly, by analyzing the posterior draws for the variance it can be shown whether the built-in volatility measure of HBART can give any economic interpretation of the volatility of share prices, which is of great importance in the financial sector.

The main results show that the machine learning methods are able to outperform the random walk in predicting the closing price of several stocks, but this performance difference shrinks to nothing as the time horizon gets longer. In general, XGBoost gave the best forecasting performance for the stock market dataset. Also, due to the added heteroscedasticity, HBART is able to make better predictions and give a better distributional fit for the stock market dataset. Lastly, the level of transaction costs severely affects the extent to which the basic trading strategy is able to achieve excess returns compared to a buy-and-hold strategy.

The rest of this paper is structured as follows. First, the methodology is explained in Section 2. Then, the different datasets used are shown in Section 3. Subsequently, Section 4 contains the results of the analyses of the datasets. Lastly, there is a discussion and conclusion in Section 5.

2 Methodology

The following section contains the methodology used to analyze the different datasets. Section 2.1 first provides the general notation used for the methods, which are described in the rest of that section. Section 2.1.1 describes BART and HBART, thereafter RF is described in Section 2.1.2. Subsequently, a description of XGBoost is given in 2.1.3. Lastly, Section 2.1.4 provides a description of RW. The methods used for evaluation of the models are given in Section 2.2.

2.1 Methods

Consider the following general model

$$y_i = G(x_i) + u_i, \text{ for } i = 1, \ldots, N, \tag{1}$$

where y_i is the i-th dependent univariate variable, $x_i = (x_{i,1}, ..., x_{i,d})$ is the i-th d-dimensional set of covariates, $G(\cdot)$ is the mapping between the covariates and the dependent variable, N is the number of observations and u_i is the i-th error with expectation 0 and a variance that may depend on the covariates. This formulation leads to the direct forecasting equation

$$\hat{y}_i = \hat{G}(x_i), \tag{2}$$

where $\hat{G}(x_i)$ is the estimated target mean function.

A special case of the general model in equation 1 is an h-step ahead model for time-series data given by

$$y_{t+h} = G_h(x_t) + u_{t+h}, \tag{3}$$

where the subscript of $G_h(\cdot)$ specifies that there is a different model for every step ahead. Moreover, the h-step ahead change can be modelled similarly

$$\Delta y_{t+h} = y_{t+h} - y_t = G_h^{'}(x_t) + u_{t+h}^{'}, \tag{4}$$

where the superscript of $G_h^{'}(\cdot)$ and $u_{t+h}^{'}$ indicate that the model is different compared to equation 3. Even though the datasets used contain both time series and non-time series data, the general notation is used where possible.

2.1.1 (H)BART

HBART, proposed in Pratola et al. (2019), is a generalisation of BART which was proposed in Chipman et al. (2010). In HBART, the response function is assumed to be equal to the sum of a mean function and a heteroscedastic error term. It has an unknown mean function $\mathrm{E}[Y|X] = f(x_i)$ alongside an unknown variance function $\mathrm{VAR}(Y|X) = s^2(x_i)$. Both are modelled using regression trees. This leads to the following formulation of the data-generating process

$$y_i = f(x_i) + s(x_i)Z, \tag{5}$$

where $Z \sim N(0,1)$. The mean function $f(x_i)$ is modelled using an ensemble of m Bayesian regression trees. Throughout this paper 200 trees are used, as recommended in Chipman et al. (2010). Such a tree is a recursive binary tree partition that consists of interior nodes and terminal nodes. The interior nodes have split rules which are determined by the predictors and particular predictor values for which the node splits: "cutpoints". Besides, a discrete probability distribution is specified on the split variables with a value in $\{1, ..., d\}$, alongside a specified discrete probability distribution for the possible cutpoints. This structure of the j-th tree is encoded in T_j. Additionally, the terminal nodes of the j-th tree have n_j^g parameter values for the mean, which are encoded in $M_j = \{\mu_{j,1}, \ldots, \mu_{j,n_j^g}\}$. This leads to the following additive regression tree model for the mean function

$$f(x_i) = \sum_{j=1}^{m} g(x_i; T_j, M_j), \tag{6}$$

where the function $g(x; T_j; M_j)$ maps the input x to a particular parameter in M_j. For the bottom nodes a normal mean prior is specified

$$\pi(\mu_{jk}) \sim N(0, \tau^2). \tag{7}$$

Since m trees are added to form the mean function $f(x)$, the prior for μ_{jk} implies the prior $f(x) \sim N(0, m\tau^2)$. The parameter τ is set by assigning a high probability to the interval of observations, i.e.

$$\tau = \frac{y_{max} - y_{min}}{2\sqrt{m}\kappa}. \tag{8}$$

With κ, the bias-variance trade-off can be altered. A higher κ leads to a smoother $f(x_i)$ with a greater probability for the mean function to account for the range of observed data, implying a smaller variance than for a lower κ. While Chipman et al. (2010) recommend $\kappa = 2$ for BART, Pratola et al. (2019) recommend a higher value like $\kappa = 5$ or $\kappa = 10$ for HBART. Unless stated otherwise, $\kappa = 2$ is used for BART and $\kappa = 5$ is used for HBART.

Furthermore, the variance function $s^2(x_i)$ is modelled using a multiplicative regression tree, consisting of $m^{'}$ trees. Throughout this paper, 40 trees are used, as recommended in Pratola et al. (2019). Comparable to the mean function, $T_l^{'}$ encodes the j-th tree structure of the variance. Likewise, the n_l^h terminal node parameters of the l-th tree for the variance are encoded in $M_l^{'} = \{s^2_{l,1}, \ldots, s^2_{l,n_l^h}\}$. This leads to the following multiplicative regression tree for the variance function

$$s^2(x_i) = \prod_{l=1}^{m'} h(x_i; T_l^{'}, M_l^{'}), \tag{9}$$

where the function $h(x; T_j^{'}; M_j^{'})$ maps the input x to a particular parameter in $M_j^{'}$. For the variance component of every tree the following prior is specified

$$s^2_{lk} \sim \chi^{-2}(\nu^{'}, \lambda^{'}), \tag{10}$$

where $\nu^{'}$ and $\lambda^{'}$ are shape parameters and $\chi^{-2}(\nu, \lambda)$ denotes the distribution $(\nu\lambda)/\chi^2_\nu$.
Equation 10 leads to the following prior for the variance function

$$s(x_i)^2 \sim \prod_{l=1}^{m'} s_l^2, \text{ with } s_l^2 \sim \chi^{-2}(\nu^{'}, \lambda^{'}). \tag{11}$$

The prior for the variance can be made equal to the prior in the homoscedastic case by matching the prior means, which leads to

$$\lambda^{'} = \lambda^{\frac{1}{m'}}, \quad \nu^{'} = \frac{2}{1 - (1 - \frac{2}{\nu})^{1/m'}}. \tag{12}$$

Equation 6 and equation 9 for the mean and variance respectively lead to the following factorisation of the posterior for the trees of the HBART model

$$\pi(T, M, T^{'}, M^{'}|y, X) \propto L(y|T, M, T', M', X) \prod_{j=1}^{m} \pi(T_j)\pi(M_j|T_j) \prod_{l=1}^{m^{'}} \pi(T_l^{'})\pi(M_l^{'}|T_l^{'}) \tag{13}$$

where

$$\pi(M_j|T_j) = \prod_{k=1}^{n^g} \pi(\mu_{jk}) \tag{14}$$

and

$$\pi(M_l^{'}|T_l^{'}) = \prod_{k=1}^{n^h} \pi(s_{lk}^2). \tag{15}$$

The full heteroscedastic regression tree is fitted using a Gibbs sampler, a Markov Chain Monte Carlo (MCMC) algorithm. In this algorithm (T_j, M_j) and $(T_j^{'}, M_j^{'})$ are drawn conditional on all other parameters and the data. For every iteration, two sets of draws are made for both the additive and the multiplicative trees. Firstly, for the m additive trees, $T_j|\cdot$ is drawn using a Metropolis-Hastings algorithm as specified in Chipman et al. (2010). Then $M_j|T_j$ is drawn from its full conditional distribution. Secondly, $T_j^{'}|\cdot$ and $M_j^{'}|T_j^{'}$ are drawn in a similar way for the $m^{'}$ multiplicative trees. This is repeated for N_{MCMC} iterations, of which the first N_{burn} are thrown away to ensure convergence, thus leaving N_{keep} posterior draws for the tree structures and terminal node parameters for the mean and variance functions. Throughout this paper, N_{MCMC} is set to 3000, of which the last 2000 draws are kept.

The posterior draws of the heteroscedastic regression tree can be used for fitting in-sample and predicting out-of-sample. Given x_i, the tree can give a posterior draw for the mean and standard deviation for every posterior draw of $(T_j, M_j, T_j^{'}, M_j^{'})$. The sequence of N_{keep} posterior draws for the mean is then given by

$$f_k^*(x_i) = \sum_{j=1}^{m} g(x_i; T_{k,j}^*; M_{k,j}^*) \text{ , for } k = \{1, \ldots, N_{keep}\}, \tag{16}$$

where $T_{k,j}^*$ and $M_{k,j}^*$ are the k-th kept posterior draw of T_j and M_j respectively. Likewise, the sequence of N_{keep} posterior draws for the standard deviation is given by

$$s_k^*(x_i) = \sqrt{\sum_{j=1}^{m^{'}} h(x_i; T_{k,j}^{'*}; M_{k,j}^{'*})} \text{ , for } k = \{1, \ldots, N_{keep}\}, \tag{17}$$

where $T_{k,j}^{'*}$ and $M_{k,j}^{'*}$ are the k-th kept posterior draw of $T_j^{'}$ and $M_j^{'}$ respectively.

A straightforward way to get a point prediction from the N_{keep} posterior draws for the mean and the standard deviation is to take the median or the average of the draws. To illustrate, taking the average leads to the following point predictions for the mean and standard deviation

$$\hat{f}(x_i) = \frac{1}{N_{keep}} \sum_{k=1}^{N_{keep}} f_k^*(x_i) \tag{18}$$

and

$$\hat{s}(x_i) = \frac{1}{N_{keep}} \sum_{k=1}^{N_{keep}} s_k^*(x_i). \tag{19}$$

For the rest of this paper, the average is used unless stated otherwise. Then, using the data-generating process specified in equation 5 we can obtain the forecast

$$\hat{y}_i = \mathrm{E}[\hat{f}(x_i) + \hat{s}(x_i)Z] = \hat{f}(x_i), \tag{20}$$

since the expectation of $Z \sim N(1, 0)$ is 0.

Under the formulation of HBART, BART is a special case of HBART with the number of multiplicative trees equal to 1. Therefore both BART and HBART can be implemented with the `rbart` R package.

2.1.2 RF

The tree-based, nonparametric RF estimation method was introduced in Breiman (2001). RF uses bagging, introduced in Breiman (1996), in order to decrease the variance of the regression trees. There are B bootstrap samples that each have a corresponding regression tree with K_b number of regions with a random selection of variables ($b \in \{1, ..., B\}$). A standard number of random regression trees is 500, which is also used throughout this paper. For every bootstrap sample we then get the following regression model

$$y_i = \sum_{k=1}^{K} c_k I_k(x_i; \theta_k), \tag{21}$$

where c_k is the average of the dependent variables that occur in the k-th region. The indicator function in equation 21 is defined as

$$I_k(x_i; \theta_k) = \begin{cases} 1, & \text{if } x_i \in R_k(\theta_k) \\ 0, & \text{otherwise,} \end{cases} \tag{22}$$

where $R_k(\theta_k)$ is the k-th region, determined by the parameter set θ_k. The regions are partitioned in order to minimize the sum of squared errors of the model. Subsequently, the forecast of the RF model is the average of the forecasts of the bootstrap samples

$$\hat{y}_i = \frac{1}{B} \sum_{b=1}^{B} \left[\sum_{k=1}^{K_b} \hat{c}_{k,b} I_{k,b}(x_i; \hat{\theta}_{k,b}) \right]. \tag{23}$$

The model is implemented using the `randomForest` R package

2.1.3 XGBoost

XGBoost is a tree boosting method introduced in Chen and Guestrin (2016). XGBoost is a modification of the gradient boosting method proposed by Friedman (2001). The idea of a gradient boost function is to sequentially refit the gradient of the loss function using small trees at each iteration (Masini et al., 2023). For a quadratic loss function, which is used in this paper, the gradient equals the residuals from the previous iteration. XGBoost speeds this process up by using greedy algorithms to determine the split rules (Chen & Guestrin, 2016). The final fitted value is computed as follows

$$\hat{y}_i = \bar{y} + \eta \sum_{m=1}^{M} f_m(x_i), \tag{24}$$

where η is the learning rate, f_m is the m-th sequential tree and $\bar{y}$ is the in-sample average of y. Hyperparameters determine how the regression trees are formed and the learning rate of the model. A number of these hyperparameters are determined using a 5-fold grid, which can be seen in Table 1. The model is implemented using the `xgboost` R package

Table 1: XGBoost: hyperparameters

Variable	Grid	Description
`nrounds`	(5,10,50,100,150)	Number of boosting rounds.
`max_depth`	(2,5,10,15)	Max tree depth.
`eta`	(0.01, 0.05, 0.1, 0.2)	Learning rate.
`gamma`	(0, 0.1, 0.2)	Min loss reduction for further partition.
`colsample`	(1)	Column subsample ratio.
`min_child_weight`	(1)	Min sum of instance weight in a child.

Note: Hyperparameters for XGBoost determined using a grid search.

2.1.4 RW

An RW process without drift is a time series process where the next observation only depends on the previous observation and white noise

$$y_t = y_{t-1} + u_t. \tag{25}$$

For an RW model the predicted value for the next period is equal to the observed current value, since $E[u_t]$, the expected value of the white noise, is equal to 0. Iteratively taking expectations of y_{t+1} leads to the following equation for the h-step ahead prediction

$$\hat{y}_{t+h} = y_t, \tag{26}$$

which directly leads to the prediction for the h-step ahead change

$$\widehat{\Delta y}_{t+h} = 0. \tag{27}$$

2.2 Evaluation

2.2.1 Plots

For the BART and HBART methods, two special types of plots are used to visualise the performance of the models. Firstly, there is the predictive quantile-quantile plot (qq-plot). This plot starts with a sample of observations (x_i, y_i). Then, for each x_i quantiles for y_i are computed using the predictive distribution $Y|x_i$. Subsequently, the quantiles of a correct model should resemble the quantiles of the uniform distribution. Hence, qq-plots can be used to visualize whether the BART and HBART models are a good fit to the out-of-sample observations.

Secondly, there is the heteroscedastic-evidence plot (H-evidence plot). This plot displays the posterior intervals for $s(x_i)$ sorted by the values of $\hat{s}(x_i)$. The plot can therefore quickly show if the conditional variance is predictor-dependent, thus potentially showing evidence for heteroscedasticity.

2.2.2 Metrics

In order to evaluate whether a sample follows a normal distribution, one can use the Jarque-Bera test for normality introduced in Jarque and Bera (1987). The statistic is computed as follows

$$JB = \frac{n}{6}\big(S^2 + \frac{1}{4}(K-3)^2\big), \tag{28}$$

where the skewness $S = \hat{\mu}_3/\hat{\sigma}^3$ and the kurtosis $K = \hat{\mu}_4/\hat{\sigma}^4$, with $\hat{\mu}_3$ and $\hat{\mu}_3$ as estimates of the third and fourth central moments and the variance σ^2. Under the null hypothesis of normality the test statistic then asymptotically follows a $\chi^2(2)$ distribution. This test is used to evaluate whether the change in stock prices is normally distributed.

Several metrics exist in order to evaluate the forecasting performance of the various methods. To start, the root mean square error (RMSE) is often used. This metric equals the root of the average of the squared residual and is given as follows

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2}, \tag{29}$$

with N the number of predictions. The mean absolute error (MAE) on the other hand equals the average of the absolute error and is given as follows

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|\hat{y}_i - y_i|. \tag{30}$$

Another commonly used metric is the Pearson correlation coefficient, which measures the linear relationship between the predicted and the actual values. Perfect forecasts give a value of 1. The coefficient is calculated with the following formula

$$\rho = \frac{\sum_{i=1}^{n}(y_i - \bar{y})(\hat{y}_i - \bar{\hat{y}})}{\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}\cdot\sqrt{\sum_{i=1}^{n}(\hat{y}_i - \bar{\hat{y}})^2}}, \tag{31}$$

where $\bar{y}$ and $\bar{\hat{y}}$ are the averages of y and $\hat{y}$ respectively.

The RMSE may be inadequate to test the quality of models that aim to learn about the distribution of population distribution, like HBART (Pratola et al., 2019). The e-statistic introduced in Szekely and Rizzo (2004) provides a way to test if two samples come from the same underlying distribution. For independent random vectors $U_1, ..., U_{n_1}$ drawn from distribution F_1 and $V_1, ..., V_{n_2}$ drawn from distribution F_2, the statistic is given as follows

$$e = \frac{n_1 n_2}{n_1 + n_2} \left(\frac{2}{n_1 n_2} \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \|U_i - V_j\| - \frac{1}{n_1^2} \sum_{i=1}^{n_1} \sum_{j=1}^{n_1} \|U_i - U_j\| - \frac{1}{n_2^2} \sum_{i=1}^{n_2} \sum_{j=1}^{n_2} \|V_i - V_j\| \right). \tag{32}$$

The BART and HBART models are evaluated using the e-statistic by comparing the percentiles calculated in the predictive qq-plot to the uniform distribution.

2.2.3 Trading algorithm

To test whether the predictions made by the methods can be useful in practice, a basic trading algorithm is constructed that can be found in Algorithm 1. In short, the algorithm checks two things for the predicted change in closing price at the end of every hour. First, if the predicted increase in closing price is greater than the level of transaction costs paid to obtain a stock, stocks are bought with all available cash. Second, if the predicted decrease in closing price is larger than the transaction costs to sell a stock, all stocks are sold. The amount of stocks and cash is updated after every transaction, as is the current portfolio value. Since RW never predicts a change in the closing price, the trading algorithm effectively corresponds to a buy-and-hold strategy where the stocks are never sold. For the other models, the portfolio oscillates between having no stocks at all and being fully invested in the stock.

3 Data

3.1 Used cars

The used cars dataset used by Pratola et al. (2019) is included in the `rbart` R package. The dataset contains 1000 observations of 7 variables of used car sales between 1994 and 2013 in the United States. The in-sample dataset is made up of 600 randomly selected observations, with the remaining 400 being part of the out-of-sample dataset. Each line contains the price of a used car alongside its trim, mileage, year of construction, colour, engine displacement and whether the owner selling the car is the first owner. An overview of the variables can be seen in Table 2. `price` is used as the dependent variable, the other variables are used as covariates. Since certain variables (`trim`, `isOneOwner`, `color` and `displacement`) are categorical variables, it is important to change these to factor values.

3.2 Alcohol and fishery

The data on alcohol use in the United States and the daily catch of fishing boats in the Grand Bank fishing grounds were first used by Fernández et al. (2002) and Kenkel and Terza (2001) respectively. Both datasets are available in the supplementary material of Pratola et al. (2019).

The dataset on alcohol comes from the United States National Health Interview survey core questionnaire and special supplements. The data contains 36 variables, of which 35 are used as independent variables and show the answers to the survey. Of these variables, 33 are binary.

Algorithm 1: Trading Simulation

Input: `predictions` – $\hat{y}$ predictions of y for $t+1$;
`actual` – y actual value at $t+1$;
`transaction_cost` – as a share of stock price;
`initial_stock_price` – price of the stock at $t=1$
Output: `portfolio_values` – portfolio value at every t;

```
// Initialize
portfolio_values ← empty list;
stock_price ← initial_stock_price;
portfolio_value ← initial_num_stocks × stock_price + initial_cash;
num_stocks ← 100;
cash ← 0;
for i in 1 to length(predictions) do
    portfolio_values.insert(portfolio_value);
    // At end of period i
    if predictions[i] > transaction_cost × stock_price and cash > 0 then
        num_buy ← cash/(stock_price × (1 + transaction_cost));
        cash ← cash − num_buy × stock_price × (1 + transaction_cost);
        num_stocks ← num_stocks + num_buy;
    end
    if predictions[i] < −1× transaction_cost × stock_price and num_stocks > 0
     then
        num_sell ← num_stocks;
        num_stocks ← num_stocks − num_sell;
        cash ← cash + num_sell × stock_price × (1− transaction_cost) ;
    end
    // During period i + 1
    stock_price ← stock_price + actual[i];
    portfolio_value ← num_stocks × stock_price + cash;
end
return portfolio_values;
```

The dependent variable is the number of alcoholic beverages consumed in the last two weeks. There are 2462 surveys included in the dataset. The in-sample dataset contains 1477 randomly selected observations, the remaining 985 are in the out-of-sample dataset.

The fishery dataset has 25 variables, 18 of them being binary. Again, one variable is used as the dependent variable. That variable is the daily catch of fishing boats in the Grand Bank fishing grounds, while the covariates variables capture the time, location and characteristics of the boat. The dataset contains 6806 observations, of which 4084 random observations are in the in-sample dataset.

3.3 Stock market

The stock market dataset contains price, volume and time data of some of the largest US stocks in the S&P 500. The change in closing price is modelled and predicted. The freely usable Alpha Vantage stock market API allows for the extraction of intraday open, close, high and low price

Table 2: Used cars dataset: overview of the variables.

Variable	Range	Description
`price`	$(995, 79995)$	Second hand price of the vehicle.
`trim`	430, 500, 550, other	Higher trim correspons to higher end vehicle.
`isOneOwner`	true, false	Has vehicle had a single owner.
`mileage`	$(1997, 255419)$	Miles that the vehicle has driven.
`year`	$(1994, 2013)$	Year that the vehicle was built.
`color`	black, white, silver, other	Colour of vehicle.
`displacement`	4.6, 5.5, other	Larger displacement corresponds to more powerful engine.

data alongside volume and time data that goes two years back for practically all US symbols: the first data point used is on June 7th 2021, while the last data point used is on May 26th 2023. The data is available in 1, 5, 15, 30 and 60-minute intervals. For example, $MSFT has 7915 observations for the hourly intervals and 30584 observations for the 15-minute interval. In this paper, only the 60-minute data is used. The data contains observations both during market hours (9:30 AM until 4:00 PM) and during the pre and post-market. The data can be extracted using the `alphavantager` R package. (H)BART, RF and XGBoost give better forecasting performance with stationary data, hence first differenced data is used for the open, close, high and low prices.

An overview of the ticker symbols that are considered can be seen in Table 3. These symbols are the eleven largest components of the S&P 500 excluding $BRK.B and $GOOG, which is identical to $GOOGL but without voting rights. $SPY is also included since that symbol tracks the entire S&P 500.

Table 3: Stock market dataset: overview of ticker symbols studied

Symbol	Full name	Description
AAPL	Apple Inc.	Consumer electronics and software
MSFT	Microsoft Corporation	Software, hardware, and services
AMZN	Amazon.com Inc.	E-commerce and cloud computing
NVDA	NVIDIA Corporation	Graphics processing units (GPUs)
GOOGL	Alphabet Inc.	Internet-related products and services
META	Meta Platforms Inc.	Social media and technology
TSLA	Tesla Inc.	Electric vehicle and clean energy
UNH	UnitedHealth Group Inc.	Health insurance and services
XOM	Exxon Mobil Corporation	Oil and gas exploration
SPY	SPDR S&P 500 ETF Trust	S&P 500 stock market index

The data received from the API is split into monthly datasets, which are joined together. Also, unless otherwise stated 5 lags are added for the change in open, close, low and high prices Additionally, a column is added for the time of the day in minutes. All in all, per stock there are 31 independent variables, including the various lags. The dependent variable that is predicted is the h-hour ahead change in closing price. The data is split into an in-sample and an out-of-sample dataset, with a roughly 50/50 split.

4 Results

The next section gives the results of the analyses of the four datasets. Section 4.1 contains the analysis of the used cars dataset. The analysis of the fishery dataset is found in Section 4.2, which is followed by the analysis of the alcohol dataset in Section 4.3. Finally, the analysis of the stock market dataset, which is the main focus of this paper, is shown in Section 4.4. For all datasets, the models are created on the in-sample dataset and subsequently evaluated for their out-of-sample predictive performance.

The analysis of the used cars, fishery and alcohol datasets are also found in Pratola et al. (2019). The analysis of the stock market dataset is not.

4.1 Used cars

For the used cars dataset, the price of second-hand cars is modelled and predicted using the mileage, trim, mileage, build year, colour, displacement and whether the car has had only one owner. Figure 1 shows the relationship between `mileage` and `price` and `year` and `price` for certain levels of `trim`. It can be seen that for `trim` = 550 and `trim` = other, the data shows a nonlinear relationship. Economically this means that for cars with less luxury features the relationship between mileage and price is less complex than for cars with more features.

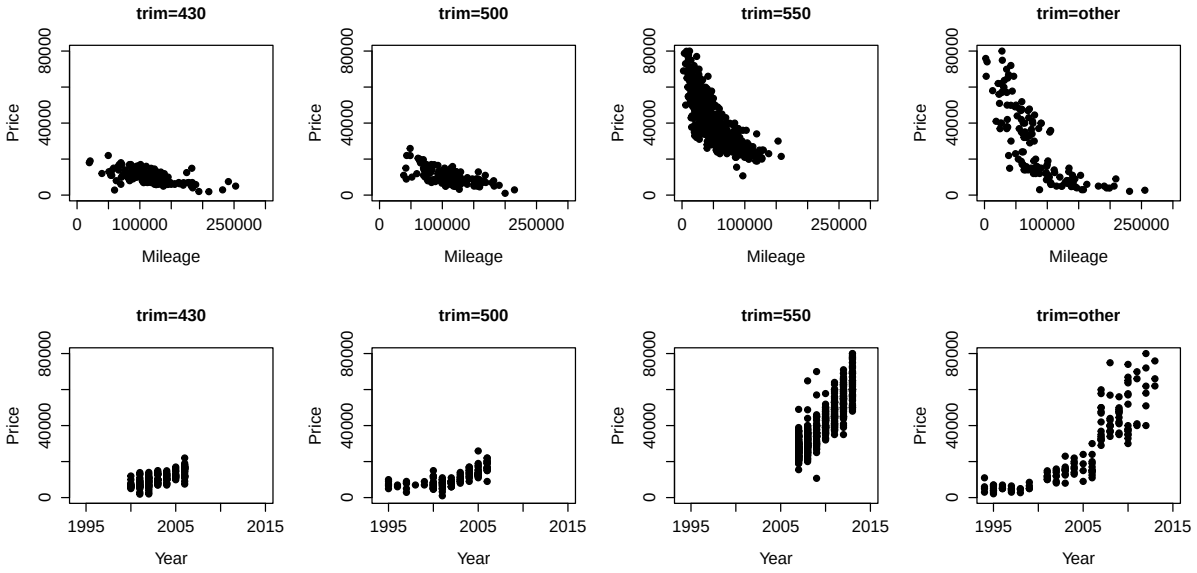

Figure 1: Used cars dataset: summary of continuous variables conditional on `trim`. The top row plots `price` versus `mileage` for the four levels of `trim`. The bottom row plots `price` versus `year` for the four levels of `trim`.

The e-statistic results of an in-sample five-fold cross-validation for various plausible values of κ for BART and HBART can be seen in Table 4. A lower e-statistic means that the percentiles of a predictive qq-plot better match the quantiles of a uniform distribution, implying a better distributional fit. The e-statistic is used as suggested by Pratola et al. (2019) in order to make the model have a better fit from a distributional perspective rather than only looking at forecasting performance. These results show that, indeed, for HBART a higher value of κ seems

applicable than for BART. Additionally, it is clear that the e-statistics are on average much lower for HBART than for BART. For the rest of the analysis of this dataset, $\kappa = 1.5$ is used for HBART. While the table may suggest a value of κ for BART between 0.25 and 0.5, $\kappa = 0.75$ is used for the rest of the analysis of this dataset[1]. These optimal values for κ are lower than the values recommended in the literature of $\kappa = 2$ for BART, and κ between 5 and 10 for HBART. This implies a relatively large variance in the price of used cars since there is a relatively low probability of the mean function accounting for the range of observed data. Economically, this makes sense because the price of used cars can be very different depending on other factors than the ones included in the dataset, like brand, range and fuel efficiency.

Table 4: Used cars dataset: cross-validation for BART and HBART.

κ	0.25	0.5	1	2	5	10	20
HBART	0.42	0.37	**0.34**	**0.34**	0.36	0.37	0.64
BART	0.87	**0.85**	0.91	1.07	1.37	1.70	2.05

Note: Average e-statistic calculated from 5-fold in-sample cross-validation of HBART and BART for varied settings of the prior mean hyperparameter κ. The minimum for each model is shown in bold.

Table 5: Used cars dataset: evaluation metrics

	ρ	RMSE	MAE	e-statistic.
RF	0.99	2423.68	1385.88	-
XGBoost	0.96	5335.92	3474.91	-
HBART	0.96	5086.45	3477.31	0.26
BART	0.96	5185.68	3456.35	1.44

Note: Correlation, RMSE, MAE and e-statistic compared to the actual values of the 400 out-of-sample predictions of `price` made by HBART, BART and RF. RF and XGBoost do not have an e-statistic since they do not specify a distribution.

The correlation, RMSE, MAE and e-statistic for the out-of-sample prediction of `price` can be seen for each method in Table 5. Since this dataset does not have time-series data, RW is not used. The RMSE and MAE of RF are much lower than for the other models. Introducing heteroscedasticity only barely reduces the prediction error for HBART compared to BART. However, the predictive qq-plots in Figure 2 show that from a distributional perspective, HBART does make better predictions since the qq-plot is closer to a straight line. This is also exemplified by the lower e-statistic for HBART in Table 5.

[1]Pratola et al. (2019) did not provide the code for the used cars dataset and the result of this cross-validation seems to be quite dependent on the seed used. Hence the results in Table 4 differ from the ones obtained in Pratola et al. (2019).

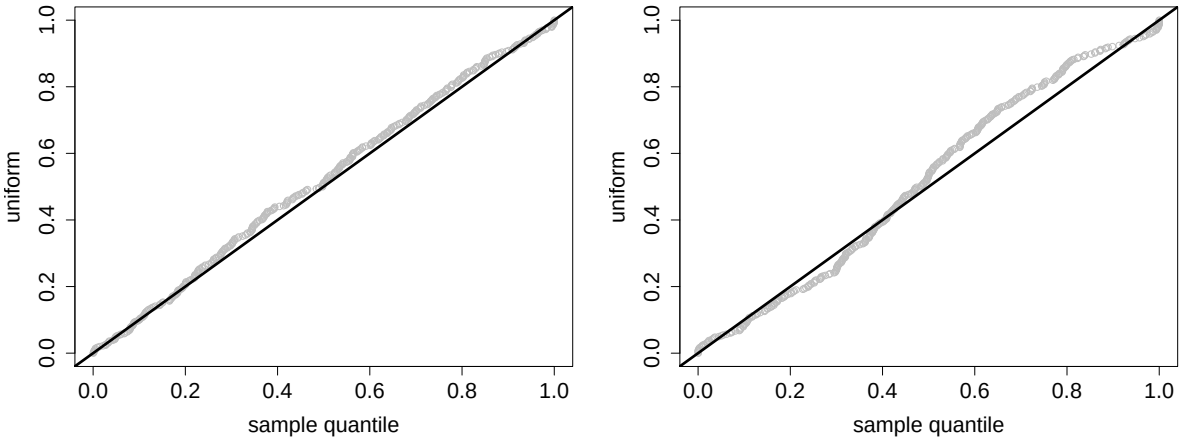

Figure 2: Used cars dataset: predictive qq-plots of posterior draws for the 400 out-of-sample predictions of `price` calibrated to the uniform distribution. Left panel: HBART with $\kappa = 1.5$. Right panel: BART with $\kappa = 0.75$

Moreover, the H-evidence plot in Figure 3 shows a presence of heteroscedasticity in the data. For very high and very low values of the posterior mean standard deviation, the estimated confidence interval using HBART of $\sigma(x)$ clearly lies outside the confidence interval implied by the BART model, even with a high degree of uncertainty in the estimate. Additionally, it can be seen that the confidence bounds of the posterior draws increase with larger $\hat{s}(x)$. As the model predicts a higher variance for a used car, the model also has a larger variance for the variance in price. Further, Figure 4 shows how the posterior samples can show the difference between `trim=other` and the other levels of `trim`. The general pattern between `mileage` and `price` is the same, but the variance is smaller for low values of `mileage` compared to other values of `trim`.

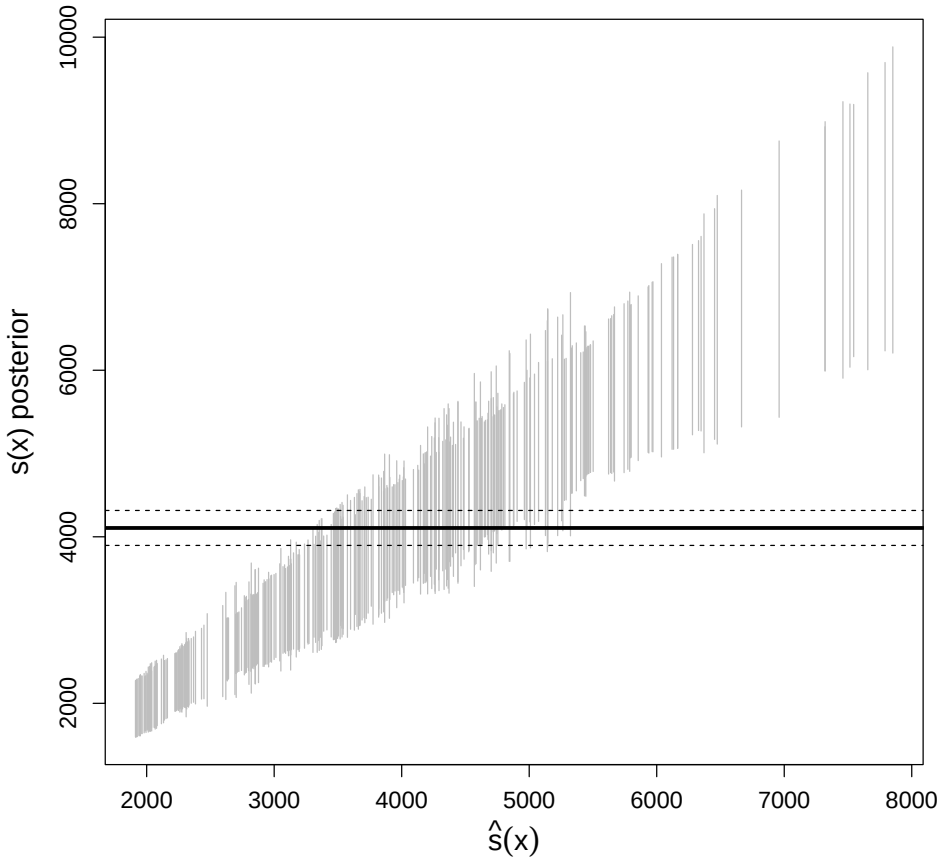

Figure 3: Used cars dataset: H-evidence plot. 90% posterior credible intervals for $s(x)$ for the HBART model versus observation index sorted by level of the posterior mean standard deviation,$\hat{s}(x)$. The solid horizontal line shows the estimate of σ from the BART model for reference, along with the 90% credible interval shown as the dashed lines.

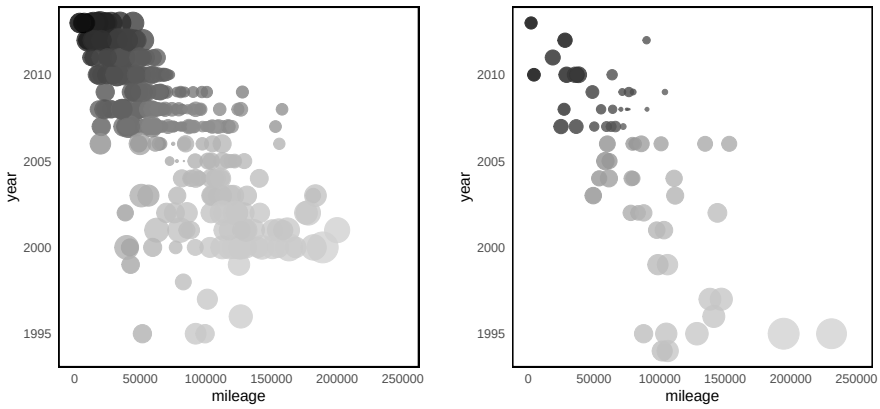

Figure 4: Used cars dataset: posterior median price for out-of-sample data plotted as a function of `year` and `mileage`. The left panel corresponds to cars with `trim.other` = 0 while the right panel corresponds to cars with `trim.other` = 1. The circles are shaded according to the posterior predicted median price with lighter denoting lower price and darker denoting higher price. Larger circles denote higher posterior predicted median standard deviation while smaller circles denote lower standard deviation.

4.2 Fishery

When considering the fishery dataset, the daily catch of fishing boats is modelled and predicted, where the covariates variables capture the time, location and characteristics of the boat. Applying the RF, XGBoost, BART and HBART methods to the fishery dataset gives evaluation metrics that can be seen in Table 6. The fishery dataset does not have time-series data, hence RW is not used. For this dataset, RF has the lowest RMSE, closely followed by XGBoost with BART and HBART further behind. However, the difference is less pronounced than for the used cars dataset in Table 5 Compared to the MAE values of the used cars dataset in Table 5, the difference between RF and (H)BART are smaller for the fishery data. This could indicate that outliers play a less prominent role in the fishery dataset. The predictive qq-plots in Figure 5 also show the better distributional fit of HBART, since the qq-plot is closer to a straight line. The e-statistic for HBART is also substantially lower for HBART than for BART. Additionally, in the left panel of Figure 5 the plug-in model $Y \sim N(\hat{f}(x), \hat{s}(x)^2)$ has been added, represented by a dashed line. This representation has an appealing simplicity and does not require knowledge of the representation of f and s to understand the output of the model. Additionally, the H-evidence plot presented in Figure 6 shows an even starker pattern than for the used cars dataset. The vast majority of confidence bounds of HBART for $s(x)$ lie outside the confidence bounds suggested by BART.

Table 6: Fishery dataset: evaluation metrics

	ρ	RMSE	MAE	e-statistic.
RF	0.76	3515.77	2129.98	-
XGBoost	0.75	3599.54	2209.92	-
HBART	0.65	4139.71	2317.28	2.73
BART	0.71	3883.22	2323.97	15.86

Note: Correlation, RMSE, MAE and e-statistic compared to the actual values of the 4084 out-of-sample predictions of the daily catch made by HBART, BART and RF. RF and XGBoost do not have an e-statistic since they do not specify a distribution.

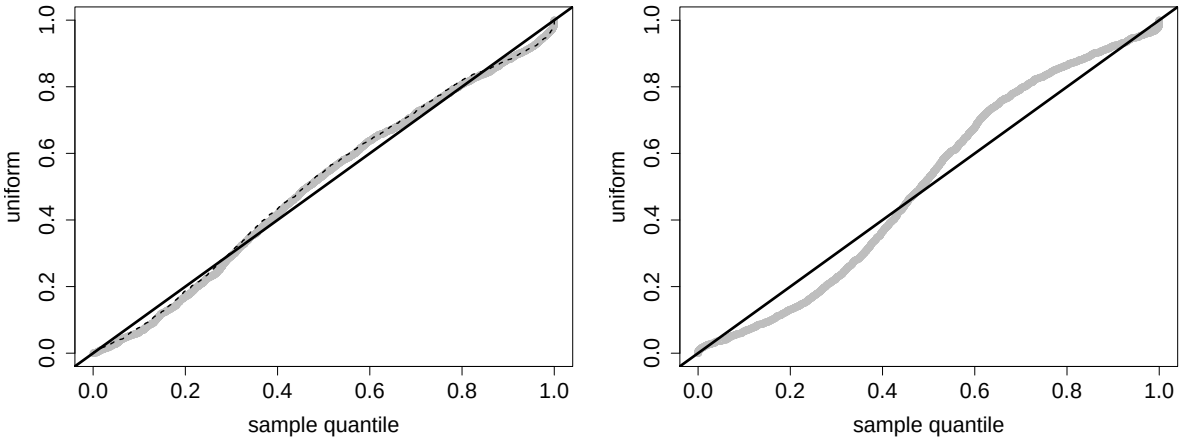

Figure 5: Fishery dataset: predictive qq-plots of posterior draws for the 2722 out-of-sample predictions of the daily catch calibrated to the uniform distribution. Left panel: HBART with $\kappa = 5$. Right panel: BART with $\kappa = 2$.

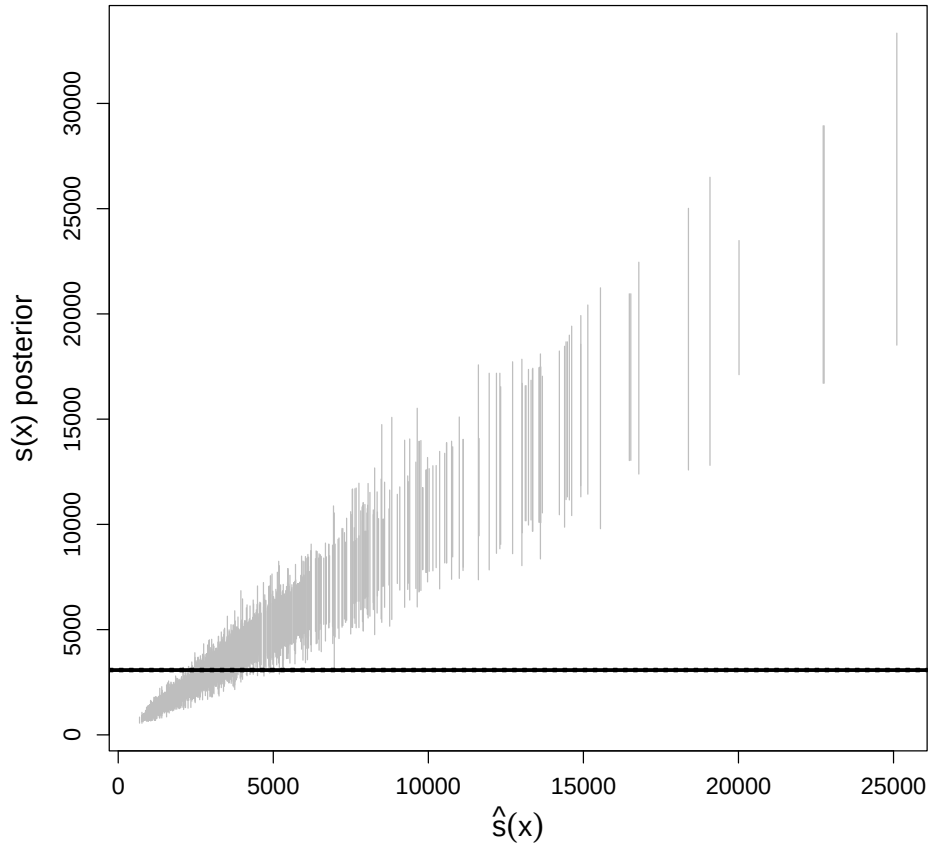

Figure 6: Fishery dataset: H-evidence plot. 90% posterior credible intervals for $s(x)$ for the HBART model versus observation index sorted by level of the posterior mean standard deviation,$\hat{s}(x)$. The solid horizontal line shows the estimate of σ from the BART model for reference, along with the 90% credible interval shown as the dashed lines.

4.3 Alcohol

In the context of the alcohol dataset, the number of alcoholic beverages consumed in the last two weeks is modelled and predicted. The covariates are the answers to a national health interview. RW is not used because this dataset does not have time-series data. Compared to the used cars and fishery datasets, the alcohol dataset paints a different picture. To start, the correlations in Table 7 of all models used are lower than they were for the other datasets in Table 5 and Table 6. This implies that the models in general are less well able to predict the dependent variable in this dataset. Besides, for this dataset BART and HBART show a lower RMSE and MAE and a higher correlation than RF does, implying a better fit. Notable is that the e-statistic is smaller for BART than it is for HBART, which implies that introducing heteroscedasticity to the BART model does not result in a better distributional fit. This can also be observed in Figure 7, where the predictive qq-plots for HBART and BART nearly look identical. The H-evidence plot in Figure 8 also differs substantially from the previous two datasets analysed: across the entire range of the posterior mean standard deviation, the 90% confidence interval of HBART falls into the 90% confidence interval of BART, further showing that for this data set heteroscedasticity does not play a large role. Apparently, the answers to questions in the questionnaire that together make up the independent variables of this dataset do not give information on the variance of alcohol use in the past two weeks.

Table 7: Alcohol dataset: evaluation metrics

	ρ	RMSE	MAE	e-statistic.
RF	0.13	1.39	1.17	-
XGBoost	0.23	1.34	1.14	-
HBART	0.24	1.34	1.14	2.67
BART	0.23	1.34	1.14	2.36

Note: Correlation, RMSE, MAE and e-statistic compared to the actual values of the 1477 out-of-sample predictions of alcohol use made by HBART, BART and RF. RF and XGBoost do not have an e-statistic since they do not specify a distribution.

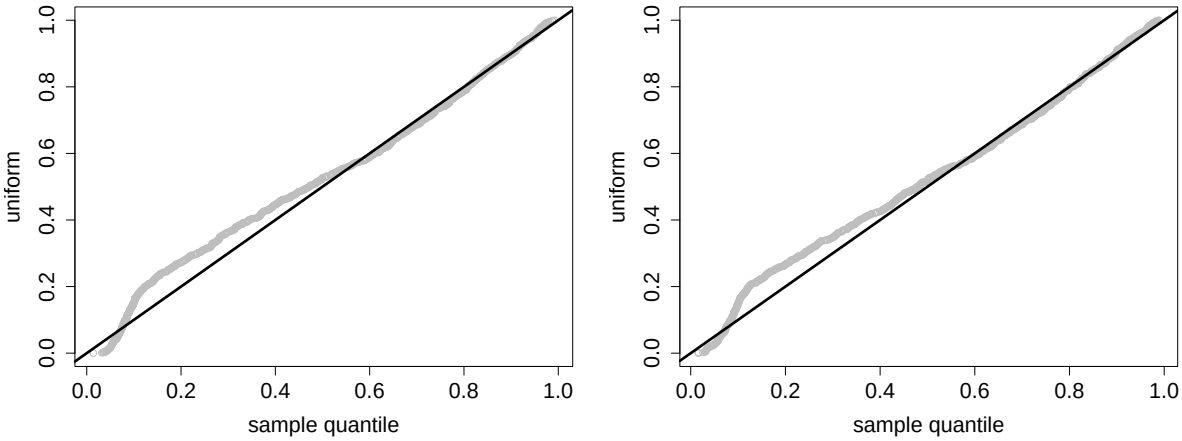

Figure 7: Alcohol dataset: predictive qq-plots of posterior draws for the 985 out-of-sample predictions of alcohol use calibrated to the uniform distribution. Left panel: HBART with $\kappa = 5$. Right panel: BART with $\kappa = 2$.

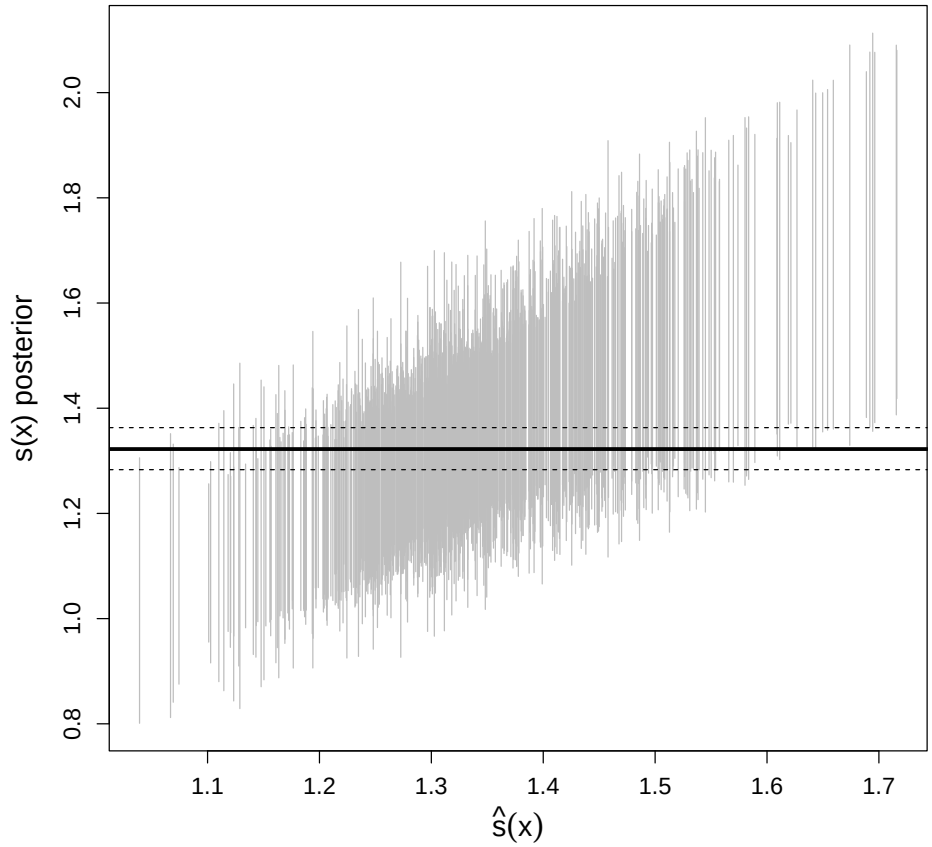

Figure 8: Alcohol dataset: H-evidence plot. 90% posterior credible intervals for $s(x)$ for the HBART model versus observation index sorted by level of the posterior mean standard deviation,$\hat{s}(x)$. The solid horizontal line shows the estimate of σ from the BART model for reference, along with the 90% credible interval shown as the dashed lines.

4.4 Stock Market

The following subsection provides an analysis of the stock market dataset. The change in the closing price of large US stocks and an S%P 500 index are modelled and predicted. Open, low, close and high prices are used as covariates, alongside volume and time data. To begin, Section 4.4.1 includes analysis on $MSFT, Microsoft Corporation stock. First, an exploratory analysis is done for the level and first differenced data. Secondly, the effect is studied of the number of lags of the open, close, low and high prices and the trading volume on the predictive performance of the models. Then, the results are given of the 5-fold in-sample cross-validation to determine κ for the BART and HBART method. Subsequently, the difference in predictive performance of the models is studied for different step-ahead forecasts. Thereafter, in Section 4.4.2 the analysis is extended to multiple ticker symbols. Lastly, the performance of the trading algorithm in Algorithm 1 using the predictions made by the HBART model is studied for various levels of transaction costs.

4.4.1 Microsoft Corporation stock

Figure 9 shows the hourly closing price of $MSFT over the entire sample, which runs from June 7th 2021 to May 26th 2023. The figure shows volatility on both the short-term and the long-term. The change in this price is predicted throughout the following analysis. Figure 10 shows a histogram of this hourly change in price. The distribution looks roughly like a bell-shaped distribution, with heavier tails than a normal distribution since there are multiple observations much farther from the mean than would be expected under a normal distribution. The descriptive statistics on the variables included in the dataset, except the time of the day, can be found in Table 8. The Jarque-Bera test rejects the null hypothesis of normality for all variables. This is due to a combination of non-zero skewness and a kurtosis above 3, the kurtosis of a normal distribution. The high kurtosis means that changes in stock prices can be more extreme than expected of a normal distribution. The volume traded is heavily skewed, which can also be seen by the large difference between the mean and median volume traded.

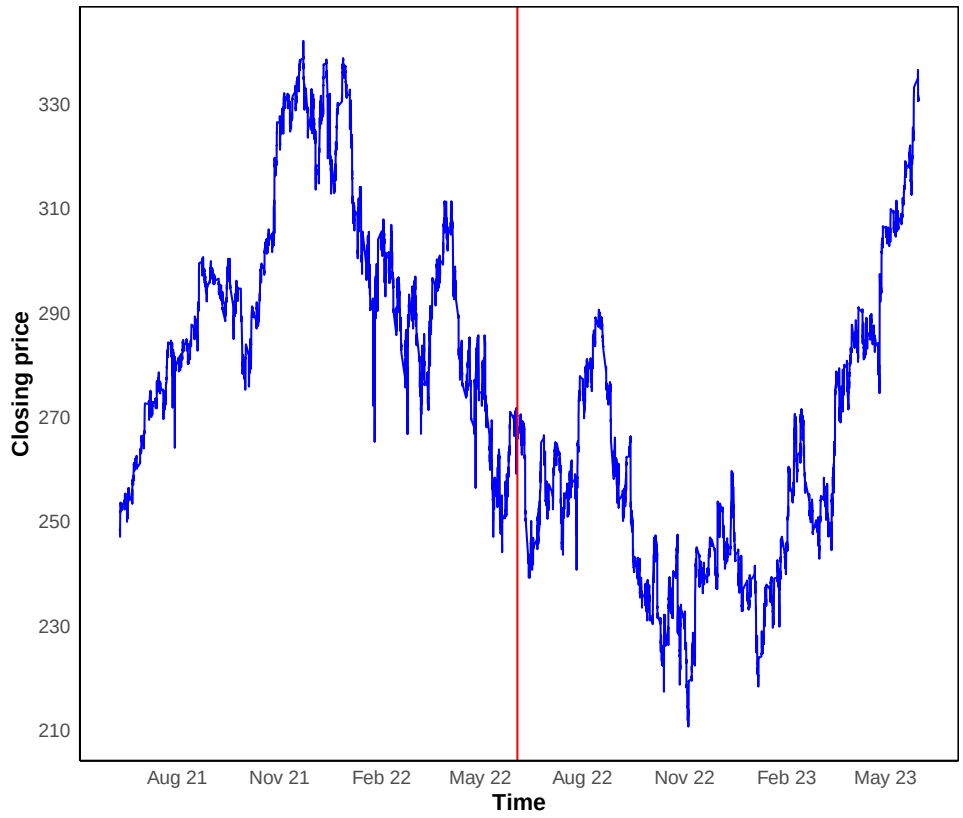

Figure 9: Stock market dataset: The hourly closing price of $MSFT over the whole sample between June 7th 2021 and May 26th 2023. The vertical line marks the cut-off point between the in-sample and out-of-sample dataset.

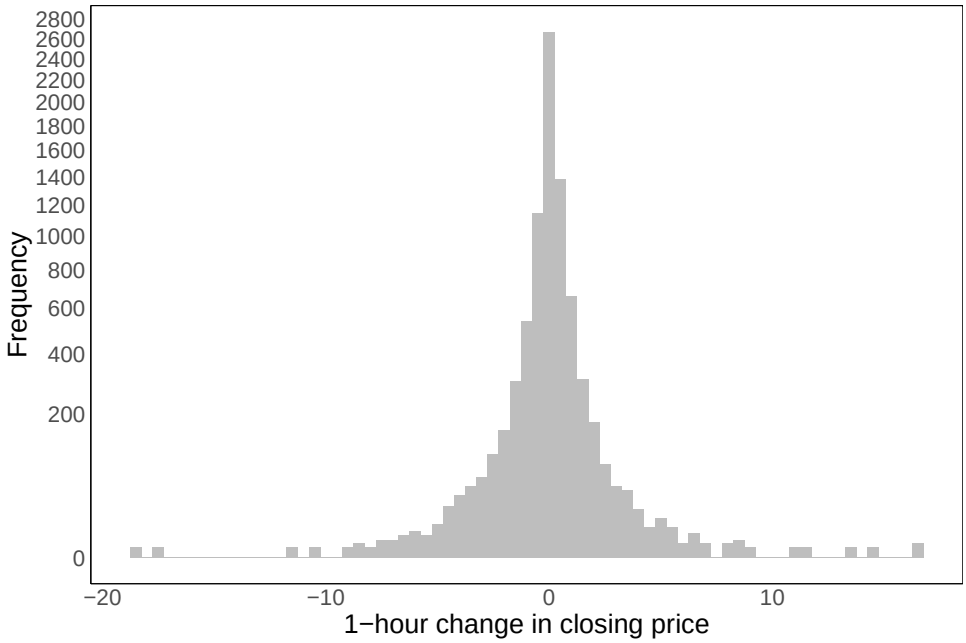

Figure 10: Stock market dataset: histogram of the 1-hour change in the closing price of $MSFT over the whole sample from June 7th 2021 to May 26th 2023. The frequency is on a square root scale.

Table 8: Stock market dataset: statistics of $MSFT data

	mean	median	σ	skewness	kurtosis	Jarque-Bera
Δ Open	0.01	0.01	1.29	0.43	15.14	75924.19
Δ Close	0.01	0.04	1.35	-0.06	23.04	175206.73
Δ Low	0.01	0.00	1.27	0.51	16.98	95576.62
Δ High	0.01	0.01	1.31	0.32	14.94	73858.40
Volume	$1.59 \cdot 10^6$	$0.68 \cdot 10^6$	$2.03 \cdot 10^6$	1.77	5.15	12876.08

Note: mean, median, standard deviation, skewness, kurtosis and Jarque-Bera statistic of the volume and hourly change in open, close, low and high price of $MSFT for the whole sample between June 7th 2021 and May 26th 2023.

Table 9 shows the RMSEs of the various models when out-of-sample predicting the hourly change in the closing price of $MSFT, using various numbers of lags for the prices and volume. Since RW does not use any predictors to make its predictions, the RMSE is identical for every number of lags. The main result that can be seen is that increasing the number of lags does not improve predictive performance for any model. This means that past prices far back into the future do not provide information on the change in the closing price 1-hour ahead. Also, using $\kappa = 5$, HBART has the lowest RMSE for all numbers of lags.

Table 9: Stock market dataset: RMSEs with different number of lags

p	5	10	20	30	40
RW	1.27	1.27	1.27	1.27	1.27
RF	83	83	83	85	85
XGBoost	81	81	82	86	83
BART	86	88	87	91	93
HBART	80	80	81	81	82

Note: RMSE compared to the actual values of the ∼3950 out-of-sample predictions of the 1-hour ahead change in the closing price of $MSFT made by RW, RF, XGBoost, BART ($\kappa = 2$) and HBART ($\kappa = 5$), using p lags of the open, close, low and high prices and the trading volume. Values for the methods except RW are given as a percentage of the RMSE of RW.

The results of the 5-fold in-sample cross-validation for BART and HBART can be found in Table 10. As for the used cars dataset, the e-statistic is used to ensure a better fit from a distributional perspective. For BART, the optimal level of κ is 2, which coincides with the recommended value by Chipman et al. (2010). Hence, $\kappa = 2$ was chosen for BART for the remaining analysis. However, the results for HBART are different compared to the recommended range of between 5 and 10 by (Pratola et al., 2019). Values under 1 give the lowest average e-statistic for this data. For the rest of the analysis, $\kappa = 0.5$ was chosen for HBART, due to the negligible difference between $\kappa = 0.25$ and $\kappa = 0.5$ and a less extreme κ may give better

forecasting performance.

The low optimal value of κ for HBART means that there is only a small probability that the mean function of HBART accounts for the range of observed data, which implies a large variance in the stock market data studied. Since the optimal value of κ for BART is higher, introducing heteroscedasticity to the model allows HBART to more successfully model the intraday volatility in the stock market.

Table 10: Stock market dataset: cross-validation for BART and HBART

κ	0.25	0.5	1	2	5	10	20
HBART	2.27	2.28	2.93	2.66	3.16	3.43	3.43
BART	4.05	4.30	4.13	3.92	7.07	10.11	13.04

Note: Average e-statistic calculated from 5-fold in-sample cross-validation of HBART and BART for varied settings of the prior mean hyperparameter κ. The minimum for each model is shown in bold.

Table 11 shows the RMSEs for the methods when forecasting out-of-sample 1, 3, 6, 9 and 12-hours ahead. The results show that the RMSE of all methods increases as the time horizon increases. This is logical, since the uncertainty of the change in closing price increases for larger time horizons. Moreover, while RW is beaten substantially by all other methods for the 1-hour ahead forecasts, this result almost entirely disappears for the longer horizons. Thus it seems that a successful trading strategy only has to include short-term predictions of the change in price. Economically this could mean that it takes time for market prices to reflect all information, meaning that in the short-run past prices provide more information on future prices than in the long run. Additionally, it can be seen that XGBoost obtains the lowest RMSE for every hour ahead, indicating a robust top performance. Also, while HBART outperforms BART for the 1 and 3-hours ahead predictions, the reverse is true for 9 and 12-hours ahead predictions: heteroscedasticity plays a smaller role in the stock market for the change in closing price over longer time horizons.

Table 11: Stock market dataset: RMSEs for different step-ahead forecasts

h	1	3	6	9	12
RW	1.27	2.26	3.15	3.86	4.43
RF	82	93	97	98	100
XGBoost	81	92	97	97	98
BART	87	95	98	98	99
HBART	85	93	98	99	100

Note: RMSE compared to the actual values of the ∼3950 out-of-sample predictions of the change in the h-hour ahead closing price of $MSFT made by RW, RF, XGBoost, BART ($\kappa = 2$) and HBART ($\kappa = 0.5$). Values for the methods except RW are given as a percentage of the RMSE of RW.

In Figure 11 one can see the predictive qq-plots of BART and HBART respectively. These plots show evidence of heteroscedasticity in the stock market data since the qq-plot of HBART more closely resembles the 45° line than the qq-plot of BART. This heteroscedasticity means that the volatility in the market is not constant and can at least partly be adequately modelled by HBART. For example, the level of price changes in the near past could affect the volatility of price changes in the future. Market participants may be drawn to act relatively quickly after large price changes, thus driving up volatility in the future.

A benefit of HBART over the other models is that HBART allows for inference of both the mean and the variance. Such an example can be seen in Figure 12: the posterior median draw shows a different pattern depending on the time of the day. On average the median posterior draw is higher around the opening of the market at 9:30 AM. The same is true for the times around the closing of the market at 4:00 PM. Also, the posterior median draw is higher at the latest time that post-market trades happen, 8:00 PM. One possible explanation for this phenomenon is that the volume traded during these hours is higher, which drives up the volatility of the price.

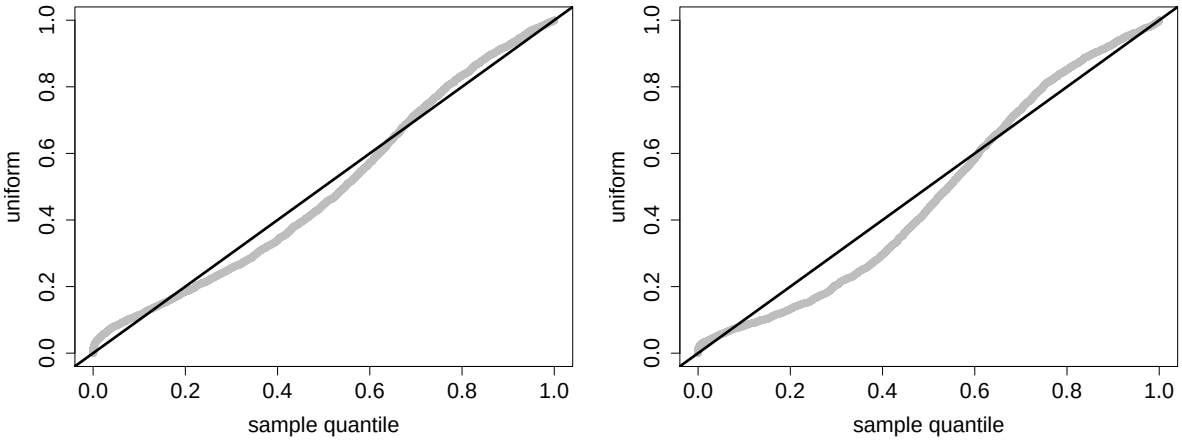

Figure 11: Stock market dataset: predictive qq-plots of posterior draws for the 3959 out-of-sample 1-hour ahead changes in the closing price of $MSFT calibrated to the uniform distribution. Left panel: HBART with $\kappa = 0.5$. Right panel: BART with $\kappa = 2$.

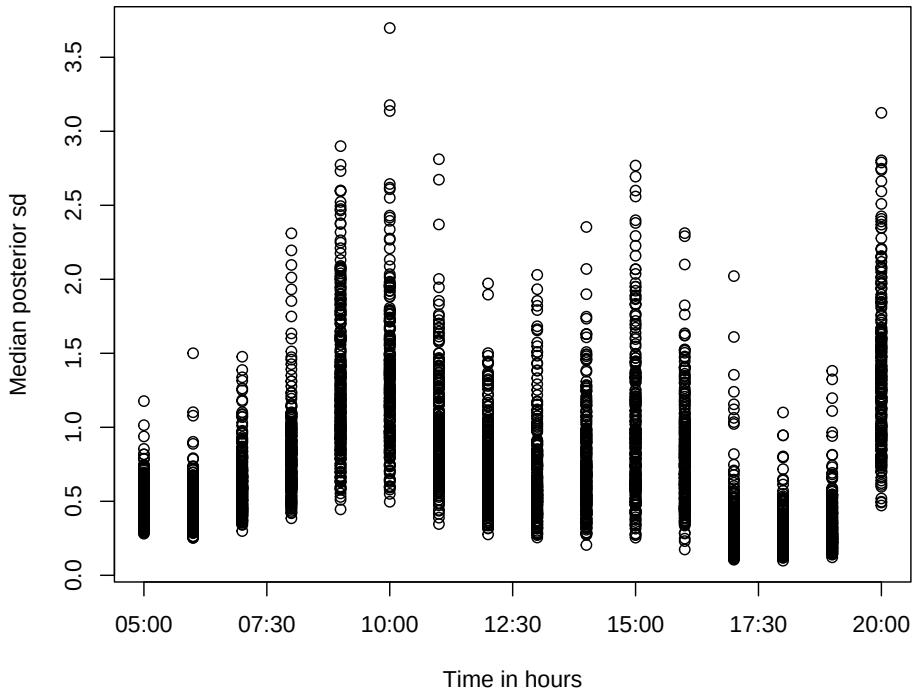

Figure 12: Stock market dataset: the median posterior draw of the standard deviation from HBART ($\kappa = 0.5$) for predicting the 3959 out-of-sample 1-hour ahead changes in closing the price of $MSFT against the time of the day in hours.

4.4.2 All ticker symbols

The degree to which the change in the closing price of stocks can be predicted differs per stock. Table 12 shows the RSMEs obtained by all methods when out-of-sample predicting the 1-hour

ahead change in closing price for some of the largest stocks in the S&P 500 index, alongside an exchange-traded tracker of that index. It can be seen that in general XGBoost obtains the lowest value for the RMSE, but it should be noted that BART and HBART were cross-validated with respect to the e-statistic, while XGBoost was cross-validated to get the lowest RMSE. The forecasting performance of the models with the exception of RW lies relatively close to each other. However, RW is outperformed by every model for every ticker symbol.

The fact that the RMSE of the RW model differs per stock means that there is a difference in volatility between the stock prices. After all, a stock with a constant price would be perfectly predictable by the RW model. The higher the average magnitude of the change in price, the higher the error of the RW model. The two stocks with the highest RMSE of the RW model are \$NVDA and \$TSLA two volatile stocks of tech companies. While the other models can outperform RW, the difference in RMSE between forecasting different ticker symbols is high for all models. The RMSEs in general are closer to the RMSE of the RW model than to the RMSE of the same model for a different symbol. In other words, the forecasting performance of all models is hampered by higher volatility.

Table 12: Stock market dataset: RMSEs for forecasting change in the closing price of all ticker symbols

Symbol	AAPL	MSFT	AMZN	NVDA	GOOGL	META	TSLA	UNH	XOM	SPY
RW	0.75	1.25	0.84	1.86	0.56	1.47	1.92	2.48	0.58	1.29
RF	84	83	88	87	89	90	84	83	89	85
XGBoost	85	82	88	86	86	89	89	83	90	84
BART	86	88	93	88	93	116	103	83	90	85
HBART	85	85	89	85	98	91	91	82	88	83

Note: RMSE compared to the actual values of the ∼3950 out-of-sample predictions of the change in the 1-hour ahead closing price of several ticker symbols made by RW, RF, XGBoost, BART ($\kappa = 2$) and HBART ($\kappa = 0.5$). Values for the methods except RW are given as a percentage of the RMSE of RW.

The performance of the trading algorithm in Algorithm 1 with a transaction cost of 0.5% of the closing price using the predictions of the models, which can be found in Table 13, shows that the algorithm gives wildly different results for each symbol. For example, \$NVDA gives exceptionally high returns, while the returns for \$GOOGL and \$UNH are relatively low for all models. An explanation could be that certain stocks are traded more frequently, thus changing prices in the short-run more extremely. It is interesting to note that ticker symbols with a generally high RMSE in Table 12 can still have a good trading performance. \$NVDA and \$TSLA have the second and third-highest RMSE, but the best trading performance in general. However, \$UNH has bad trading performance and a high RMSE.

Table 13: Stock market dataset: trading performance with 0.5% transaction costs for all ticker symbols

Symbol	AAPL	MSFT	AMZN	NVDA	GOOGL	META	TSLA	UNH	XOM	SPY
RW	18.92	22.59	12.41	106.32	9.00	63.56	-19.45	-0.01	12.39	2.74
RF	43.87	27.47	**88.92**	**465.77**	-2.74	72.01	**441.99**	6.85	26.90	12.38
XGBoost	13.85	41.51	57.77	398.45	**24.09**	**163.96**	285.91	-3.77	31.39	**22.00**
BART	**95.91**	5.42	44.19	360.15	9.84	-71.12	26.52	**24.06**	**37.63**	4.50
HBART	25.43	**52.33**	55.60	278.64	1.82	85.32	78.65	0.16	29.26	-2.85

Note: Returns in percentage achieved with the ∼3950 out-of-sample predictions of the 1-hour ahead change in the closing price of several ticker symbols made by RW, RF, XGBoost, BART ($\kappa = 2$) and HBART ($\kappa = 0.5$) using a basic trading algorithm with a transaction cost per trade of 0.5% of the share price. The highest returns for every symbol are given in bold.

The level of transaction costs has a profound influence on trading performance with Algorithm 1. The excess returns of the algorithm using the 1-hour ahead out-of-sample predictions by HBART compared to a buy-and-hold strategy can be seen in Table 14. These excess returns are equal to the return of the returns of the algorithm minus the returns of a buy-and-hold strategy. The results clearly show that without transaction costs it would be possible to obtain immense profits. An explanation for this fact is that market participants do not buy and sell stocks all the time due to the presence of transaction costs. Hence, past prices still have predictive power for prices in the future. In general, the higher the transaction costs, the lower the profits. This does not hold every time for the highest transaction costs studied: sometimes a transaction cost of 2% gives a better return than for 1%. This is caused by the fact that some trades that turn out badly aren't done with a higher transaction cost, since for a trade to be done by the algorithm the change in predicted closing price has to be larger with a transaction cost of 2% than for 1%.

Table 14: Stock market dataset: trading performance of HBART for different levels of transaction costs

%	0	0.01	0.1	0.2	0.5	0.75	1.0	2.0
AAPL	3637.68	3101.23	729.23	168.97	6.51	-6.31	-10.20	0.00
MSFT	3538.53	2894.78	717.97	202.03	29.74	14.06	-10.14	-7.81
AMZN	4232.56	3573.38	1494.19	640.36	43.19	-7.83	-15.20	9.03
NVDA	61420.14	51749.26	13468.27	4134.22	172.32	78.23	75.08	-31.83
GOOGL	2013.47	1654.45	618.00	159.04	-7.18	-9.51	6.61	-20.23
META	1644.38	1419.94	411.26	104.25	21.76	-4.56	0.30	0.00
TSLA	47854.35	40146.30	9858.78	2381.80	98.10	-25.05	-39.20	-37.94
UNH	1143.97	964.52	341.17	110.71	0.17	-9.05	1.83	0.00
XOM	3212.67	2695.68	608.23	170.36	16.87	9.82	3.04	0.00
SPY	1161.42	915.78	172.41	52.25	-5.60	6.00	-7.12	0.00

Note: Excess returns compared to a buy-and-hold strategy achieved with the ∼3950 out-of-sample predictions of the 1-hour ahead change in the closing price of several ticker symbols made by HBART ($\kappa = 0.5$) using a basic trading algorithm for various percentages of the share price as transaction costs.

5 Conclusion

This study compared the performance of a random walk and the random forest, XGBoost, BART and HBART machine learning methods for forecasting the intraday change in stock prices in a technical analysis setting. The change in closing price was predicted using the time of the day, (lagged) values of the volume and change in open, close, low and high prices. The results of the stock market dataset show that the machine learning methods can outperform the random walk in forecasting the change in the stock prices studied. However, this performance difference decreases substantially as the time horizon gets longer, with almost no difference for a 12-hours ahead forecast. Generally, XGBoost gave the best forecasting performance with regard to the RMSE. By introducing heteroscedasticity to the BART model, HBART can give better forecasts than BART and is able to adequately model the heteroscedastic variance during the day. From a distributional perspective, the model also gives a better fit. The forecasts were also evaluated using the performance of a forecast-based trading algorithm, which showed that transaction costs greatly affect the degree to which profits can be made using the algorithm. Generally, for transaction costs above 0.5% the strategy does not perform well enough using the predictions made by HBART to achieve excess returns compared to a buy-and-hold strategy. The forecasting and trading performance also differ substantially per stock.

Additionally, analysis was done for predicting the prices of used cars, alcohol use and the number of fish caught, using the random forest, XGBoost, BART and HBART methods. These results show that by introducing heteroscedasticity the distributional fit of the predictions made by HBART is improved compared to BART for the used cars and fishery dataset. This result does not hold for the alcohol dataset. However, the forecasting performance of HBART with regards to the RMSE is not improved substantially in these datasets compared to BART. For the used cars and fishery dataset, random forest gives the best forecasting performance. However, for the alcohol dataset random forest gives the worst performance.

Further research can extend the stock market analysis to other financial products such as bonds, commodities, foreign exchange, small caps and options. Also, higher frequency data can be used than hourly data. Besides, the reason for the difference in forecasting performance for different stocks can be studied. Including fundamental data as predictors may further improve forecasting performance too.

References

Ahmed, S., Alshater, M. M., Ammari, A. E., & Hammami, H. (2022). Artificial intelligence and machine learning in finance: A bibliometric review. *Research in International Business and Finance*, *61*, 101646.

Athey, S., & Imbens, G. W. (2019). Machine learning methods that economists should know about. *Annual Review of Economics*, *11*(1), 685–725.

Ayala, J., García-Torres, M., Noguera, J. L. V., Gómez-Vela, F., & Divina, F. (2021). Technical analysis strategy optimization using a machine learning approach in stock market indices. *Knowledge-Based Systems*, *225*, 107119.

Basak, S., Kar, S., Saha, S., Khaidem, L., & Dey, S. R. (2019). Predicting the direction of stock market prices using tree-based classifiers. *The North American Journal of Economics and Finance*, *47*, 552–567.

Breiman, L. (1996). Bagging predictors. *Machine learning*, *24*, 123–140.

Breiman, L. (2001). Random Forests. *Machine Learning*, *45*, 5–32.

Caginalp, G., & DeSantis, M. (2011). Nonlinearity in the dynamics of financial markets. *Nonlinear Analysis: Real World Applications*, *12*(2), 1140–1151.

Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794.

Chipman, H. A., George, E. I., & McCulloch, R. E. (2010). BART: Bayesian additive regression trees. *The Annals of Applied Statistics*, *4*(1).

Fama, E. F. (1970). Efficient capital markets: A review of theory and empirical work. *The Journal of Finance*, *25*(2), 383.

Fama, E. F. (1995). Random walks in stock market prices. *Financial Analysts Journal*, *51*(1), 75–80.

Fernández, C., Ley, E., & Steel, M. F. J. (2002). Bayesian modelling of catch in a north-west atlantic fishery. *Journal of the Royal Statistical Society Series C: Applied Statistics*, *51*(3), 257–280.

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, *29*(5).

Jarque, C. M., & Bera, A. K. (1987). A test for normality of observations and regression residuals. *International Statistical Review / Revue Internationale de Statistique*, *55*(2), 163.

Kenkel, D. S., & Terza, J. V. (2001). The effect of physician advice on alcohol consumption: Count regression with an endogenous treatment effect. *Journal of Applied Econometrics*, *16*(2), 165–184.

Lakhchini, W., Wahabi, R., & El Kabbouri, M. (2022). Artificial intelligence machine learning in finance: A literature review. *International Journal of Accounting, Finance, Auditing, Management and Economics*, *3*(6-1), 437–455.

Lanbouri, Z., & Achchab, S. (2020). Stock market prediction on high frequency data using long-short term memory. *Procedia Computer Science*, *175*, 603–608.

Masini, R. P., Medeiros, M. C., & Mendes, E. F. (2023). Machine learning advances for time series forecasting. *Journal of Economic Surveys*, *37*(1), 76–111.

Medeiros, M. C., Vasconcelos, G. F. R., Veiga, Á., & Zilberman, E. (2021). Forecasting Inflation in a Data-Rich Environment: The Benefits of Machine Learning Methods. *Journal of Business & Economic Statistics*, *39*(1), 98–119.

Menkhoff, L. (2010). The use of technical analysis by fund managers: International evidence. *Journal of Banking &amp Finance*, *34*(11), 2573–2586.

Nishiyama, K. (1998). Some evidence on regime shifts in international stock markets. *Managerial Finance*, *24*(4), 30–55.

Pratola, M. T., Chipman, H. A., George, E. I., & McCulloch, R. E. (2019). Heteroscedastic BART via multiplicative regression trees. *Journal of Computational and Graphical Statistics*, *29*(2), 405–417.

Szekely, G., & Rizzo, M. (2004). Testing for equal distributions in high dimension. *InterStat*, *5*.

A Source code

For replication purposes, all code and datasets used in this study are available in the depository under the supplementary material. All code is written in R version 4.1.0 using RStudio version 2023.03.0. Further documentation can be found in the general README.md file, which describes the structure of the depository.

The scripts included in the depository are:

- `alcohol.R`: script that analyzes the alcohol dataset;
- `e_stats.R`: script that can get the e-statistic for BART and HBART, and can cross-validate these models with regards to that statistic;
- `fishery.R`: script that analyzes the fishery dataset;
- `loading_data.R`: script that allows for loading the stock market dataset;
- `predict_with_xgboost.R`: script that can make an XGBoost model from data;
- `qinsamp.R`: script that gets the quantiles for BART and HBART that are necessary to compute the e-statistic;
- `simulate_trading.R`: script that can simulate trading performance using stock market predictions;
- `stock_market.R`: script that analyzes the stock market dataset;
- `used_cars.R`: script that analyzes the used cars dataset.

The analyses of the used cars, fishery, alcohol and stock market datasets are performed by running their respective scripts from top to bottom. A more extensive description is found in the README.md file in the depository under /src. Figures are stored automatically as pdfs in the depository, and the tables are automatically stored in the environment.

The R packages used are: `alphavantager`, `coda`, `caret`, `dplyr`, `e1071`, `energy`, `forecast`, `ggplot2`, `gridExtra`, `lubridate`, `Metrics`, `randomForest`, `rbart`, `scales`, `tseries`, `xgboost`, `xtable` and `zoo`