# Prediction of Default Probabilities in the Context of Credit Risk Stress Testing: A Comparison of Methods

**Master's Thesis**

in the study program Business Mathematics
at the Department of Mathematics and Computer Science

**Philipps-Universität Marburg**

submitted by
Maximilian Hennig
on 18 June 2025

supervised by
Prof. Dr. Hajo Holzmann

# Danksagung

An dieser Stelle möchte ich die Gelegenheit nutzen, mich bei allen zu bedanken, die mich bei der Erstellung dieser Masterarbeit unterstützt und begleitet haben.

Mein besonderer Dank gilt meinem Betreuer, Professor Dr. Hajo Holzmann, für seine kontinuierliche Hilfsbereitschaft bei organisatorischen Fragen in der Vorbereitungsphase der Arbeit. Die regelmäßige Rücksprache während des Schreibprozesses sowie die Unterstützung bei inhaltlichen Fragen und der Literaturbeschaffung habe ich stets als sehr zuverlässig und hilfreich empfunden.

Darüber hinaus möchte ich Dr. Vanessa Dräger, Simon Niederauer und Lukas Weber von der Deutschen Bundesbank für ihre Beratung bei der Themenwahl und für die Bereitstellung relevanter Daten danken – ebenso für die interessanten inhaltlichen Impulse, die sich aus den mehrfachen Gesprächen ergeben haben. Ich freue mich sehr, dass wir diese Zusammenarbeit letztlich realisieren konnten.

# Contents


**1 Introduction** **1**

**2 Forecast Design** **3**

2.1 Data . . . 3

2.1.1 Endogenous Variable of Interest . . . 3

2.1.2 Macroeconomic Indicators . . . 4

2.2 Training and Testing . . . 5

2.3 Hyperparameter Tuning . . . 7

**3 Methods** **10**

3.1 Modified Bayesian Model Averaging . . . 10

3.2 Bayesian Model Averaging . . . 14

3.3 Approximating the Posterior Model Probability . . . 15

3.4 Elastic Net . . . 18

3.5 Tree Based Models . . . 20

3.5.1 Bayesian Additive Regression Trees . . . 21

3.5.2 Extreme Gradient Boosting . . . 24

**4 Forecast Results** **28**

4.1 Consistent Scoring Functions . . . 28

4.2 Weighting by Time Period . . . 32

4.3 Description of Results . . . 34

4.3.1 Forecasts . . . 34

4.3.2 Scores . . . 36

4.3.3 Boxplots . . . 39

4.3.4 Diebold-Mariano-Test . . . 42

4.3.5 Feature Importance . . . 43

**5 Conclusion** **46**

**A Forecast Graphics** **48**

**B Scores** **54**

**C Boxplots** **60**

**D Results of the Diebold-Mariano Test** **66**

# 1 Introduction

*Credit risk stress testing* is one of the main aspects of macrofinancial supervisory stress tests. Stress testing originated in the 1990s as part of banks' internal risk management, when they wanted to estimate the effects of a possible negative trend in price development. As Siemsen and Vilsmeier (2017) point out, **supervisory** stress tests were introduced after the 2008 banking crisis and in the following years. These tests aim to determine commercial banks' capital needs and ensure financial stability. The European Central Bank (ECB) conducted a substantial test in 2014 to ensure the adequate capitalization of the largest banks in the euro area. Nowadays, national central banks in Europe periodically execute supervisory stress tests (since 2015 in Germany) to evaluate the capital reserves of smaller institutions as well.
The purpose of credit risk stress tests is to estimate the *Expected Loss* (EL) of credits given to debtors by the bank. This is done for a "normal" baseline scenario and an adverse scenario, in which a strong recession is simulated using different macroeconomic indicators. The expected loss depends on the *Default Probability* (PD), which is the probability that a debtor will default within a given time horizon; the *Loss Given Default* (LGD); and the *Exposure at Default* (EaD), which is calculated using the following equation: $\mathrm{EL}_t = \mathrm{LGD}_t \cdot \mathrm{PD}_t \cdot \mathrm{EaD}_t$, $t$ denoting a point in time. A baseline or adverse scenario is typically twelve quarters long. This research focuses on forecasting the PD time series for this period using multiple macroeconomic indicators and past PD values. The question arises as to which explanatory variables are appropriate for the dataset. This problem is widely referred to as *model uncertainty*.

According to Clyde and George (2004), methods using Bayesian statistics to address model uncertainty have evolved remarkably over the last few decades. As early as 1978, Leamer (1978) worked on regression selection based on prior beliefs, where he proposed different methods of sequentially dropping variables. *Bayesian Model Averaging* (BMA) was first described by Raftery (1995) and forms the basis of the current benchmark method for credit risk stress testing. However, regression-based models are not the only useful approach. Chipman et al. (1998) were the first to use a Bayesian approach to tree models to solve the problem of variable selection. Since then, this approach has led to the development of several successful models. Guth (2022) used multiple estimation methods, such as shrinkage, regression, tree models, and neural networks, on a dataset of expected default frequencies and macroeconomic variables for Austria to compare their prediction accuracy and robustness. As far as we know, this was the first comparison of methods in a credit risk stress testing context.
In this study, we perform a similar comparison of methods for the German banking sector, dividing debtors into seven economic sectors, resulting in seven independent models per method and seven comparisons. We analyze three regression-based and two tree-based methods that deal with model uncertainty differently in detail. *Bayesian Model Averaging* (BMA) uses a large number of small regression models and combines them into a single model using a weighted average based on posterior model probabilities. *Modified Bayesian Model Averaging* is quite similar, but it applies multiple filters to combine only a subset of econometrically and economically plausible models. This method is the closest to the so-called *Benchmark Constraint Bayesian Model*

*Averaging* (BCBMA) that is currently used in German stress tests, as described by Siemsen and Vilsmeier (2017). Additionally, BCBMA creates a one-factor model by mapping macroeconomic conditions to a single systemic factor using quantile mapping. *Elastic Net*, developed by Zou and Hastie (2005), is a combination of Lasso and Ridge regression. That means, an Elastic Net is based on a linear regression model, but it shrinks the model parameters using an additional constraint that can set parameters to zero, thereby performing variable selection. *Bayesian Additive Regression Trees* (BART), introduced by Chipman et al. (2010), are based on a combination of a large number of relatively small regression trees. A posterior distribution is placed over the space of possible tree structures, from which new candidates are sampled. The model prediction is obtained by averaging the outputs of the sampled ensemble. In contrast, *Extreme Gradient Boosting* (XGBoost), developed by Chen and Guestrin (2016), builds the model in a sequential manner. At each iteration, a new regression tree is added to the existing ensemble, selected to most effectively reduce the remaining error of the model. Additionally, we compare the estimations of these models with those of two much simpler methods that do not use any exogenous variables to get an idea of how helpful complex models are in general for estimating PDs.

This research work is constructed as follows: The dataset is introduced in Section 2, including a detailed description of the sectors and macroeconomic indicators. This section also outlines the necessary time series transformations required for reliable estimation. The subsequent part covers the training and forecasting procedures, along with hyperparameter tuning for each model. Section 3 presents the theoretical background of the methods under consideration, including a mathematical explanation of how they address model uncertainty and generate forecasts. Forecasting results are discussed in Section 4, where the performance of the different models is compared. This section also provides a brief overview of consistent scoring functions and introduces the four evaluation metrics used. Particular attention is given to forecasts during periods of crisis, as estimated probabilities of default (PDs) under adverse scenarios have direct implications for a bank's required capital reserves. Section 5 concludes with a discussion of these results and recommendations concerning a possible switch in the current forecasting method.

# 2 Forecast Design

In this section, we first provide a detailed explanation of the default probabilities and the macroeconomic indicators that we use to explain them. Then, we discuss the data used for the training process and the estimated values. Additionally, we use a similar algorithm to tune the models' hyperparameters.

## 2.1 Data

The data to which the methods that will be analyzed are applied consists of the dependent variable *Probability of Default (PD)* and various macroeconomic indicators as explanatory variables. All data are available as quarterly data, i. e. they were measured at the end of March, June, September and December of each year. We describe the data in more detail below.

### 2.1.1 Endogenous Variable of Interest

Institutions must report loans of € 1 million (€ 1.5 million until 2015) or more to an individual borrower or borrower unit at the end of each quarter. The Deutsche Bundesbank collects these loans in the *Bundesbank's credit register* dataset.[1] Since 2008, this dataset has also included the probability of default for each loan, which is based on the Internal Ratings-Based (IRB) approach. This approach requires institutions to estimate the probability that a borrower will default within the next twelve months. These estimates rely on historical data and statistical models, and they are validated by banking supervision.
For each economic sector, the PDs at the loan level are weighted by loan volume and averaged. The Deutsche Bundesbank extracted Information on volumes, PDs and sectors at the loan level from the *Bundesbank's credit register* and provides the aggregated data for this research project. The seven sectors that will be analyzed here are:

- *Sovereigns*: Nations with rating class BB or better (Sov)
- *High-Risk Sovereigns*: Nations with rating class B or worse (Sov-HR)
- *Financial institutes* (Financial)
- *Non Financial Corporates*: not secured by mortgage (NFC-nonRE)
- *Households*: not secured by mortgage (HH-nonRE)
- *Non Financial Corporates*: Collateral in the form of commercial real estate (NFC-RE)
- *Households*: Collateral in the form of residential real estate (HH-RE)

For each sector there is a time series with quarterly data from March 2008 to December 2022 ($t = 1, \ldots, 60$). The methods are compared separately for each time series, which may lead to different results regarding the predictive quality of the methods depending on the sector.

[1] For more information see https://www.bundesbank.de/de/aufgaben/bankenaufsicht/einzelaspekte/gross-und-millionenkredite/ueberwachung-des-kreditgeschaefts-hinsichtlich-gross-und-millionenkredite-598938

Assuming we have decided on one of the seven sectors, we have the time series $PD_1, \ldots, PD_{60}$ of default probabilities, where $PD_1$ corresponds to the PD from March 2008 and $PD_{60}$ to that from December 2022. To obtain the endogenous variable, which we will use in the models, we apply the following transformations.

**Definition 2.1** (**Delta-Logit-PD**)**.** The time series of *Logit-PDs* is defined by

$$z_t := \log\left(\frac{PD_t}{1 - PD_t}\right), \qquad t = 1, \ldots, 60. \tag{2.1}$$

The time series of *Delta-Logit-PDs* is defined by

$$y_t := \Delta z_t = z_t - z_{t-4} = \log\left(\frac{PD_t}{1 - PD_t}\right) - \log\left(\frac{PD_{t-4}}{1 - PD_{t-4}}\right), \qquad t = 5, \ldots, 60, \tag{2.2}$$

where $\Delta$ denotes the year-on-year change.

*Remark* 2.1. The logit transformation (2.1) guarantees that our predictions of the PDs represent interpretable probabilities after the back-transformation with

$$\widehat{PD} = \frac{1}{1 + \exp(-\hat{z})}. \tag{2.3}$$

Since the exponential function only returns positive values, $\widehat{PD}$ will only take values in the interval $(0, 1)$.
The second transformation, i. e. the $\Delta$-transformation in (2.2), ensures the stationarity of the time series that will be fitted to the models. To validate the stationarity of the Delta-Logit-PDs we use the *Augmented Dickey-Fuller* test.

### 2.1.2 Macroeconomic Indicators

For the choice of exogenous variables, we follow Siemsen and Vilsmeier (2018) and Guth (2022). The following macroeconomic indicators are also available on a quarterly basis between 2000 and 2022. Their values are shown in Figure 1.

- Construction Price Index (*CHPI*): An average value of indices for office buildings and industrial buildings, including turnover tax; available from the Federal Statistical Office of Germany - Consumer prices → Construction price indices → Residential buildings and non-residential buildings on https://www.destatis.de/EN/Service/OpenData/short-term-indicators.html#461634

- Residential Property Price Index (*RHPI*): An index for purchase prices of new and existing apartments and houses; provided in the ECB's *Statistical Data Warehouse* data portal - https://data.ecb.europa.eu/data/datasets/RESR/RESR.Q.DE._T.N._TR.TVAL.4D0.TB.N.IX

- Unemployment Rate (*UNEMP*): For persons aged from 15 to 74; also available in the *Statistical Data Warehouse* - https://data.ecb.europa.eu/data/datasets/IESS/IESS.Q.DE.S.UNEHRT.TOTAL0.15_74.T

- Long-Term Interest Rate (*BONDS*): Loans with a maturity of ten years are used, the counterpart sector is not specified; also available in the *Statistical Data Warehouse* - https://data.ecb.europa.eu/data/datasets/IRS/IRS.M.DE.L.L40.CI.0000.EUR.N.Z
- Short-term Interest Rate (*SWAP1Y*): We use the Euribor for a term of twelve months; also available in the *Statistical Data Warehouse* - https://data.ecb.europa.eu/data/datasets/FM/FM.M.U2.EUR.RT.MM.EURIBOR1YD_.HSTA
- Gross Domestic Product for Germany (*GDP*): seasonally adjusted; published by the Federal Reserve Bank of St. Louis - https://fred.stlouisfed.org/graph/?g=X0vl
- Consumer Price Index (*CPI*): For non-food and non-energy items, seasonally unadjusted; also provided by the Federal Reserve Bank of St. Louis - https://fred.stlouisfed.org/graph/?g=X2c4
- Gross Domestic Product for the United States (*USGDP*): seasonally adjusted; also provided by the Federal Reserve Bank of St. Louis - https://fred.stlouisfed.org/graph/?g=X0AP
- Gross Domestic Product and its Main Components for the 27 EU Countries (*EUGDP*): seasonally adjusted; provided by *EUROSTAT* - https://ec.europa.eu/eurostat/databrowser/view/naidq_10_gdp/default/table?lang=de
- DAX Performance Index (*DAX*): Closing prices on the last day of each quarter; available at *Yahoo Finance* - https://finance.yahoo.com/quote/DAX/

*Remark* 2.2. As with PDs, all macro variables are differentiated before being used as exogenous variables in the model. This means that we build the difference to the previous quarter and to the same quarter from the previous year. Let $X_t$, $t = 1, \dots, 60$ be the time series of one of the macro variables. Then

$$\begin{aligned} X_t^Y &= \Delta_Y \log(X_t) = \log(X_t) - \log(X_{t-4}), \quad t = 5, \dots, 60, \quad \text{and} \\ X_t^Q &= \Delta_Q \log(X_t) = \log(X_t) - \log(X_{t-1}), \quad t = 5, \dots, 60, \end{aligned} \tag{2.4}$$

are used as explanatory variables. Note that *UNEMP* is already a rate variable and, like *BONDS* and *SWAP1Y*, can also take negative values. Therefore, we do not use the log-transformation here and instead use $X_t^Y = \Delta_Y X_t$ and $X_t^Q = \Delta_Q X_t$. As we use the quarter-on-quarter changes and the year-on-year changes of each macroeconomic variable, this results in a total of 20 explanatory variables. After this transformation, all time series except *BONDS*, *SWAP1Y*, *CPI* and *CHPI* are stationary with a probability of more than 95 percent. We only use deltas and not growth rates, as these do not lead to a higher probability of stationarity in all but two cases.

## 2.2 Training and Testing

In the context of credit risk stress testing, we can use the entire time series as a training data set. The resulting model is used to estimate default probabilities for the next three years (twelve quarters). Although we are working with time series, the estimates are not traditional forecasts.

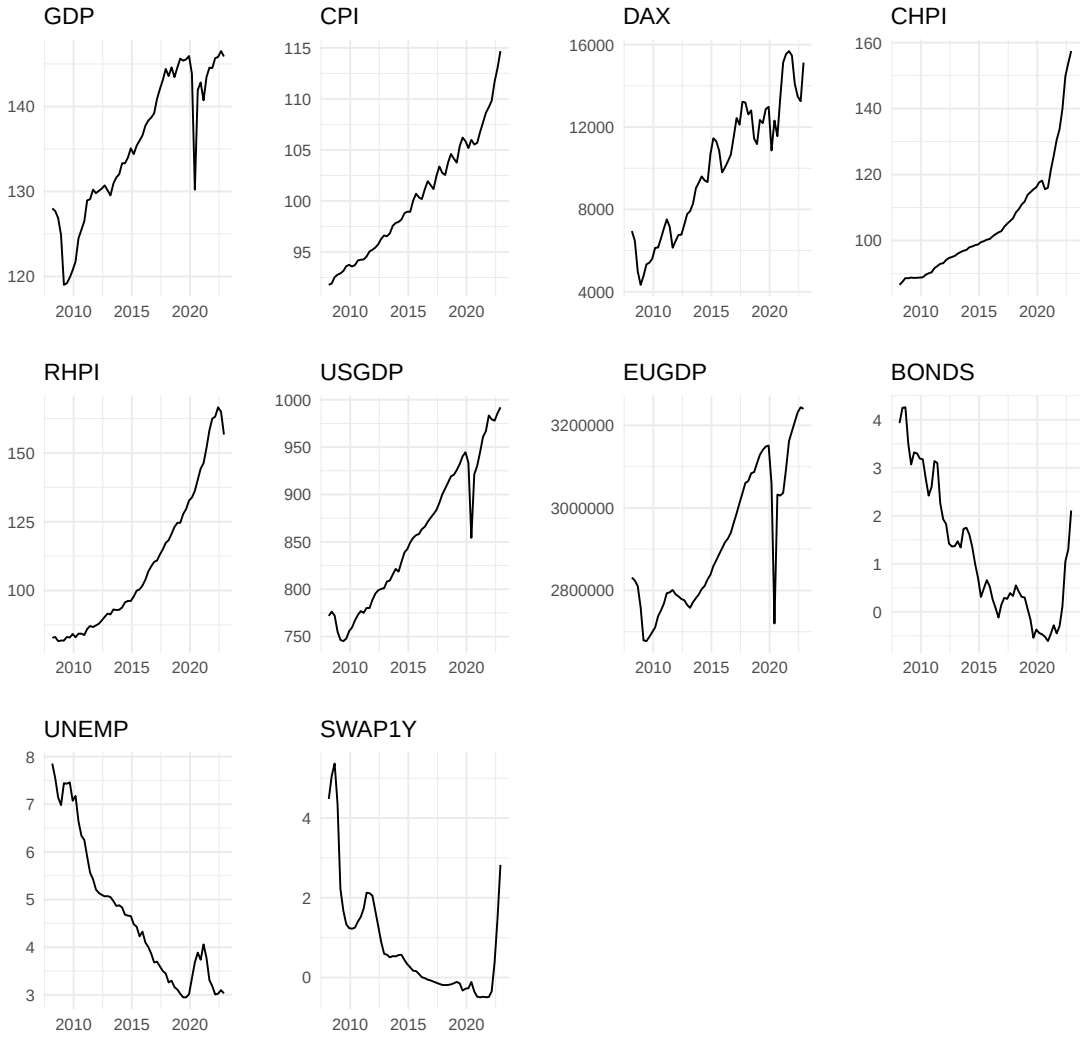


Figure 1: Macro variables between 2008Q1 and 2022Q4

In training the model, we analyze the extent to which the exogenous macro variables and the lags of the endogenous variable affect the target variable. The values of the exogenous variables are taken as given for the forecast horizon, i. e. the next 12 quarters. Consequently, the estimation procedure approaches an estimation that would be applied to no time series data. However, due to the lags of the endogenous variable, there are time series effects in the estimation. This means that the PDs must be estimated sequentially, as the current estimate may affect the subsequent ones.

In our case, we have to divide the data set into a training and a testing data set in order to be able to evaluate the out-of-sample performances of the examined methods. Analogous to the stress testing context, the testing data set always directly follows after the training data set and has a length of twelve quarters. If we were to test on only a single time interval, this could negatively influence the result of the comparison. A procedure that is actually worse could appear to be superior because the time series in the test interval happens to match that procedure well. To avoid this, we test on different intervals by varying the length of the training data set. Before defining the training and testing data sets mathematically, we reindex the

endogenous and exogenous time series such that $t = 1$ corresponds to March 2009, and $t = 56$ corresponds to December 2022.

**Definition 2.2.** Let $y = (y_1, \dots, y_{56})$ be the dependent time series resulting from (2.2) and $\mathbb{X} \in \mathbb{R}^{56\times(N+K)}$ the data matrix containing all $N$ exogenous variables as outlined in (2.4) and the $K$ lags of the endogenous variable that are used in the model. We define the *training data set* $D^{Tr}$ and the *testing data set* $D^{Te}$ in the $j$-th iteration as follows:

$$D^{Tr_j} := (y^{Tr_j}, \mathbb{X}^{Tr_j}), \quad \text{where} \quad y^{Tr_j} := (y_1, \dots, y_{j+3}), \quad \mathbb{X}^{Tr_j} := \mathbb{X}_{[1:(j+3),:]} \tag{2.5}$$

$$D^{Te_j} := (y^{Te_j}, \mathbb{X}^{Te_j}), \quad \text{where} \quad y^{Te_j} := (y_1, \dots, y_{j+15}), \quad \mathbb{X}^{Te_j} := \mathbb{X}_{[(j+4):(j+15),:]} \tag{2.6}$$

One single iteration consists of the following four steps:

1) Train model on $(y_1, \dots, y_k)$ and $\mathbb{X}_{[1:k,:]}$

2) Predict next value $\hat{y}_{k+1}$ using $\mathbb{X}_{[k+1,:]}$, the corresponding exogenous values

3) Replace $y_{k+1}$ by $\hat{y}_{k+1}$ in $\mathbb{X}$

4) Repeat 2) and 3) until twelve values in a row have been predicted

In step 2, we take the next row of $\mathbb{X}$ as given, as this corresponds to the specified scenarios in the context of stress tests. In step 3, the true values $y$ must be replaced by the estimates $\hat{y}$, as these are unknown in the application context and later estimates also depend on the previous estimates.
We follow Guth (2022) and start the first iteration with a small data set of length 4 to train our first model, i. e. $D^{Tr_1} = \big((y_1, \dots, y_4), \mathbb{X}_{[1:4,:]}\big)$. Once the model has been computed, we estimate $\hat{y}^{Te_1} = (\hat{y}_5, \dots, \hat{y}_{16})$ using the test data set $D^{Te_1}$. In particular, we use the exogenous values of the forecast horizon $\mathbb{X}_{[5:16,:]}$ and we may use lagged endogenous values $(y_1, \dots, y_4)$ depending on the model. Subsequently we evaluate our estimation using $y^{Te_1} = (y_5, \dots, y_{16})$. If $\mathbb{X}$ contains one lag of $y$, we have to carry out a stepwise estimation. At first, we calculate $\hat{y}_5$ using $\mathbb{X}_{[5,:]}$ and $y_4$, then calculate $\hat{y}_6$ using $\mathbb{X}_{[:,6]}$ and $\hat{y}_5$ and so on. After each iteration we increase the length of the training data set by one resulting in

$$D^{Tr_{41}} = \big((y_1, \dots, y_{44}), \mathbb{X}_{[1:44,:]}\big) \quad \text{and} \quad D^{Te_{41}} = \big((y_1, \dots, y_{56}), \mathbb{X}_{[45:56,:]}\big) \tag{2.7}$$

in the last iteration. Please note that – due to the limited PD time series – the training time series begins with a higher index if the model contains lags. For example, if the model contains two lags, we must start our training data at $i = 3$, since we need $y_1$ and $y_2$ as explanatory values for our first model. This procedure is precisely described below in the algorithm 1.

## 2.3 Hyperparameter Tuning

For the methods described in the next chapter, different hyperparameters must be set. These affect the training algorithm. Since they do not change dynamically during training, it is important to set them manually beforehand.

**Algorithm 2** How to train models of different lengths and estimate Delta-Logit-PDs for the next twelve quarters

$L \leftarrow$ empty list
**if** No lags in model **then**
  **for** $j = 1$ to 41 **do**
    Train *model* on $y^{Tr_j} := (y_1, \ldots, y_{j+3})$ and $\mathbb{X}^{Tr_j} := \mathbb{X}_{[1:(j+3),:]}$
    Estimate $\hat{y}^{Te_j} = (\hat{y}_{j+4}, \ldots, \hat{y}_{j+15})$ using *model* and $\mathbb{X}^{Te_j} := \mathbb{X}_{[(j+4):(j+15),:]}$
    Save $\hat{y}^{Te_j}$ in $L$ at position $j$
  **end for**
**else if** One lag in model **then**
  **for** $j = 1$ to 40 **do**
    Train *model* on $y^{Tr_j} := (y_1, \ldots, y_{j+4})$ and $\mathbb{X}^{Tr_j} := \mathbb{X}_{[2:(j+4),:]}$
      ▷ $(y_2, \ldots, y_{j+4})$ as dependent, $(y_1, \ldots, y_{j+3})$ as lagged explanatory variable
    $V \leftarrow$ empty vector
    $vol \leftarrow y_{j+4}$ (vector of lags)
    **for** $k = 1$ to 12 **do**
      Estimate $\hat{y}_{j+4+k}$ using *model* and $\mathbb{X}_{[(j+4+k),:]}$ and *vol*
      Save $\hat{y}_{j+4+k}$ in $V$ at position $k$
      $vol \leftarrow \hat{y}_{j+4+k}$
    **end for**
    Save $V$ in $L$ at position $j$
  **end for**
**else if** $l > 1$ lags in model **then**
  **for** $j = 1$ to $41 - l$ **do**
    Train *model* on $y^{Tr_j} := (y_1, \ldots, y_{j+3+l})$ and $\mathbb{X}^{Tr_j} := \mathbb{X}_{[(l+1):(j+3+l),:]}$
      ▷ $(y_{l+1}, \ldots, y_{j+3+l})$ as dependent, $(y_1, \ldots, y_{j+2+l})$ as lagged explanatory variable
    $V \leftarrow$ empty vector
    $vol \leftarrow (y_{j+6-l}, \ldots, y_{j+5})$ (vector of lags)
    **for** $k = 1$ to 12 **do**
      Estimate $\hat{y}_{j+3+k+l}$ using *model* and $\mathbb{X}_{[(j+3+k+l),:]}$ and *vol*
      Save $\hat{y}_{j+3+k+l}$ in $V$ at position $k$
      $vol \leftarrow (vol_{[2:]}, \hat{y}_{j+3+k+l})$
    **end for**
    Save $V$ in $L$ at position $j$
  **end for**
**end if**

We perform a *grid search* to find the most effective hyperparameter tuple. This involves setting possible values for each hyperparameter. Each possible combination of these values is then tested, and the combination that gives the best result is used for the final model. This procedure can be very time consuming even with just a few hyperparameters, so it is not possible to test many values per parameter. In particular, this method will not find the optimum for parameters over a continuous range of values. The hyperparameters we examine and the values we ultimately use are described in the chapters on the respective methods.
To evaluate how good a combination of hyperparameters is, we start with a training time series $t = 1, \ldots, 4$ of length four and estimate $y$ for the next twelve quarters as described above. Estimates are compared with actual values using the *Mean Squared Error* (MSE). The training time series is extended by one quarter, and the procedure is repeated until the final training time series of length 42 and which ends in 2019Q4 is reached. The arithmetic mean is calculated from the MSE values. The hyperparameter combination is evaluated on the basis of this mean.

# 3 Methods

This chapter introduces the regression- and tree-based methods selected for comparison in this study. For each approach, models are trained and their forecasts evaluated to assess the adequacy of the respective methods for the given dataset of PD time series.

## 3.1 Modified Bayesian Model Averaging

The current state-of-the-art credit risk satellite method is precisely described by Siemsen and Vilsmeier (2018). It is called *Benchmark-Contrained Bayesian Model Averaging (BCBMA)* and combines a classical *BMA* approach with a structural filter derived from a Merton/Vasicek model (benchmark model). In this model, the distributions of macro variables are fitted to past values for estimation. The value of a single systemic factor is derived from the quantiles of a macroeconomic scenario in these distributions using quantile mapping. Then, the model translates this value into a PD. If estimates of a model deviate too far from those of the benchmark model, this model is filtered out. For reasons of complexity, we do not include the benchmark filter in this research work. For further information, see Vasicek (2002).

We call our method *Modified BMA* because we incorporate three macroeconomic and statistically motivated filters into the *BMA* method. With the exception of the benchmark filter, we adhere exactly to the descriptions of Siemsen and Vilsmeier (2018). The main problem with a simple OLS model is the short observation period. The short PD time series contains hardly any observations that can be attributed to specific macroeconomic events. The explanatory variables, some of which are highly correlated, lead to serious model uncertainty. If the available exogenous variables were used in a single linear model, there would be a risk of overfitting. OLS estimates outside the observation period would be sensitive to the model specifications and therefore less reliable.

The main idea of *BMA* to deal with this problem is to generate many linear models with only a few exogenous variables each. Unsuitable models are filtered out. The remaining models are merged into a single model using a weighted average based on their posterior model probabilities. We start with an *autoregressive distributed lag* (ADL) model equation that describes the relationship between the delta logit PDs and the macroeconomic variables:

$$y_t = \alpha_0 + \sum_{k=1}^{K} \alpha_k y_{t-k} + \sum_{l=0}^{L} \beta_l' x_{t-l} + \varepsilon_t \tag{3.1}$$

$y$ is defined by (2.2), $\varepsilon$ is the residual and $K \in \mathbb{N}$ and $L \in \mathbb{N}$ denote the number of endogenous and exogenous lags. $x_t \in \mathbb{R}^{20}$ is a vector of the ten QoQ and ten YoY differences of the macro variables, i. e. the $t$-th row of the matrix $\mathbb{X}$. $\alpha_k \in \mathbb{R}$ and $\beta_l \in \mathbb{R}^{20}$ denote the model parameters that are later estimated using the OLS method, $\alpha_0$ denotes the intercept parameter. For variables that do not appear in a model, we enter zeros in the appropriate places in $\beta$ so that all parameter vectors have the same length.

We now create a separate model for each possible combination of exogenous and endogenous variables, where the maximum number of explanatory variables is $N$. Since we have a total

of 20 macroeconomic variables available, this results in $\binom{20(L+1)+K}{N}$ models, each with its own model equation (3.1). We define the model space as the set of parameters that occur in one of the models:

$$\mathcal{M} := \left\{ (\alpha_0, \alpha_1, \ldots, \alpha_K, (\beta_0)_1, \ldots, (\beta_0)_{20}, \ldots, (\beta_L)_1, \ldots, (\beta_L)_{20})_i \middle| 1 \leq i \leq \binom{20(L+1)+K}{N} \right\} \tag{3.2}$$

$(\beta_x)_y$ denotes the parameter value for macro variable $y$ of lag $x$. Each parameter vector $m_i \in \mathcal{M}$ represents a specific unique model. In order to filter out the unimportant models and identify the meaningful models on which the final model is based, we carry out the five steps described below in turn. In each step, we reduce the model space $\mathcal{M}$ using a specific filter. The indicators on which the individual filters are based are:

- Step 1 – Adjusted Coefficient of Determination
- Step 2 – Akaike Information Criterion
- Step 3 – Correlation coefficients of exogenous variables
- Step 4 – Autocorrelation of residuals
- Step 5 – Signs of Long Run Multipliers

In a sixth step, the remaining models are combined to form a final model.

**Step 1 – Leaps and bounds**
Since $\mathcal{M}$ so far consists of the models of all possible combinations of exogenous variables, it can be assumed that $\mathcal{M}$ contains some models that do not help us estimate $y$. Therefore, in a first step, we want to filter out most models by keeping only models in $\mathcal{M}$ with up to four explanatory variables. This reduces the likelihood of overfitting, especially in the case of a short training time series. Furthermore, fewer co-variables also mean fewer possibilities for collinearity. For each model size we only keep the $Q$ best models. $Q$ should be chosen sufficiently large. To rank the models, the adjusted $R^2$ is used as a benchmark. Furnival and Wilson (2000) developed an algorithm that finds these models in an efficient way without calculating the parameters and $R^2$ values of all models. The best method that works with our dataset is **regsubsets** of the R package **leaps**. We set $Q = 40$ because it is the maximum number that does not produce any errors with our relatively small dataset. At this point the filtered model space contains only 140 models, 40 models with four, three and two explanatory variables each and 20 models with only one variable.

**Step 2 – Occam's window**
At this point, we estimate all $Q$ models in $\mathcal{M}$. Madigan and Raftery (1994) use Occam's window to identify the relevant subset of the model space. To do this, they use the posterior model probabilities $p_i := p(m_i|D)$ given our data $D$. $D$ denotes the training data set which is used for model $m_i$. $p(m_i|D)$ indicates the probability that $m_i$ is the correct model and thus shows us

how well it fits our data $D$. Instead of calculating $p_i$ we follow Burnham and Anderson (2002) and approximate the posterior model probability by

$$p_i \approx \frac{\exp(-0.5\Delta_i)}{\sum_{j=1}^{Q} \exp(-0.5\Delta_j)} =: \delta_i, \tag{3.3}$$

with $\Delta_i = AIC_i - \min_{1 \leq j \leq Q}\{AIC_j\}$, $AIC$ denoting the *Akaike Information Criterion*. We will describe in more detail later why we can approximate $p_i$ by $\delta_i$. For a threshold $o \in \mathbb{R}$, we remove all models $m_i$ from the model space $\mathcal{M}$ for which

$$\frac{\max_{1 \leq j \leq Q}\{\delta_j\}}{\delta_i} > o. \tag{3.4}$$

In this way, we keep all models that are at most $o$ times less likely to be the right model than the best model.

**Step 3 – Multicollinearity filter**

We only want to consider models based on a combination of macro- variables whose pairwise correlation is below a certain threshold $\gamma \in \mathbb{R}$. Let $m_i$ be a model from the remaining model space $\mathcal{M}$. $m_i$ contains the macro variables $\overline{x}^i = \{\hat{x}^1, \ldots, \hat{x}^{n_i}\}$, i. e. variables in $\overline{x}^i$ have a non-zero parameter, $n_i$ being the number of variables in $m_i$. If $j, k \in \{1, \ldots, n_i\}$, $j \neq k$, exist such that

$$Corr(\hat{x}^j, \hat{x}^k) > \gamma \tag{3.5}$$

then $m_i$ is filtered out of $\mathcal{M}$.

**Step 4 – Autocorrelation filter**

Next, we want to filter out all models with first order autocorrelation. In this case, the value of one residual would depend significantly on the previous one. In the model equation $\varepsilon_t = \rho_1 \varepsilon_{t-1} + \nu_t$, the estimator $\hat{\rho}_1$ would be significantly different from zero. Siemsen and Vilsmeier (2018) recommend using the Durbin-Watson test. It is known that the test underestimates autocorrelation when the model is autoregressive. In our models, where there are lags in the dependent variable, the test statistic is biased. The recommendation is therefore to use the Durbin-h test, whose test statistic is not biased by autoregression. Nevertheless, we use the Durbin-Watson test to compare the described procedure.

**Step 5 – Sign restrictions**

As part of this filter, we make some assumptions about the impact of macro variables on PDs. We formulate concrete ideas about the effect of shocks to certain macro variables. If a model does not fit these assumptions, it is filtered out of $\mathcal{M}$. To do this, we exogenously impose sign restrictions $s_j$ on the estimated long-run multipliers (LRM). The LRM of a macrovariable $\hat{x}_j$ is

defined as

$$\theta_j := \frac{\sum_{l=0}^{L}(\hat{\beta}_l)_j}{1 - \sum_{k=1}^{K}\hat{\alpha}_k}. \tag{3.6}$$

If, for example, we do not use the first lag of the dependent variable in the model, then the parameter $\hat{\alpha}_1$ would be zero. Let $m_i \in \mathcal{M}$ be a model that contains the macro variables $\overline{x}^i = \{\hat{x}^1, \ldots, \hat{x}^{n_i}\}$. Then $m_i$ is filtered out if at least one $j \in \{1, \ldots, n_i\}$ exists such that

$$sgn(\theta_j) \neq s_j. \tag{3.7}$$

Like Siemsen and Vilsmeier (2017) we impose the following signs on the macro LRMs:

| Macro Variable | LRM Sign |
|---|---|
| UNEMP | + |
| BONDS | + |
| GDP | - |
| CHPI | - |
| RHPI | - |
| DAX | - |
| EUGDP | - |
| USGDP | - |
| CPI | 0 |
| SWAP1Y | 0 |

The positive sign of the LRM of *UNEMP* means that we assume that rising unemployment will lead to higher PDs in the long term. A zero means that these LRMs remain unconstrained.

**Step 6 – Model averaging**

After these five steps, we have generated the filtered model space from the unfiltered model space $\mathcal{M}$, which we denote by $\overline{\mathcal{M}}$. Let $R = \#\overline{\mathcal{M}}$ be the number of remaining models. To combine these into a weighted average, we need to recalculate their posterior model probabilities with

$$p_i \approx \frac{\exp(-0.5\Delta_i)}{\sum_{i=1}^{R}\exp(-0.5\Delta_i)} = \delta_i. \tag{3.8}$$

Now we use them as weights when calculating a single parameter vector as follows:

$$m^{BMA} = \sum_{i=1}^{R}\delta_i m_i, \quad with \quad m_i = (\alpha_0, \alpha_1, \ldots, \alpha_K, (\beta_0)_1, \ldots, (\beta_0)_{20}, \ldots, (\beta_L)_1, \ldots, (\beta_L)_{20})_i \tag{3.9}$$

We analyze the following hyperparameters:

| Para | Financial | NFC-nonRE | Sov | NFC-RE | HH-nonRE | HH-RE | Sov-HR |
|---|---|---|---|---|---|---|---|
| $o$ | 30 | 30 | 30 | 30 | 30 | 30 | 30 |
| $L$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $\gamma$ | 0.9 | 0.8 | 0.9 | 0.7 | 0.9 | 0.7 | 0.7 |
| DW | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 |

## 3.2 Bayesian Model Averaging

In addition, we will analyze the *Bayesian Model Averaging (BMA)* method and compare the results with those of the modified variant of the previous chapter. *BMA* was described by Raftery (1995) and was also implemented in the R library **BMA** which we use in this research work. The biggest difference to the modified version is that no macroeconomic filters are used here. Instead, all possible models are considered and weighted according to their posterior model probability.

The model equation of a single linear model remains (3.1) and does not change. The same 20 macro variables and possibly their lags and the lags of the dependent variable are considered. The posterior model probability for a model $m_i$ given our dataset $D$ is then approximated by

$$p_i = p(m_i|D) = \frac{\exp(-0.5BIC_i)}{\sum_{j=1}^{Q} \exp(-0.5BIC_j)} = \delta_i. \tag{3.10}$$

Then we apply Occam's window to cut off all models that are $o$ times less likely to be the right model than the likeliest one. All remaining models form the basis for the final average model.

We analyze the following hyperparameters:

| Para | Financial | NFC-nonRE | Sov | NFC-RE | HH-nonRE | HH-RE | Sov-HR |
|---|---|---|---|---|---|---|---|
| $o$ | 30 | 30 | 30 | 20 | 20 | 20 | 30 |
| $L$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

## 3.3 Approximating the Posterior Model Probability

Let us now briefly examine why the approximation to the optimal weighting of the models in (3.10) makes sense.

Let $A$ and $B$ be events and $\mathcal{P}$ a probability distribution with $\mathcal{P}(B) \neq 0$. By Bayes' theorem we know that for the conditional probability $\mathcal{P}(A|B)$, given that $B$ is true, the following equation holds:

$$\mathcal{P}(A|B) = \frac{\mathcal{P}(B|A)\mathcal{P}(A)}{\mathcal{P}(B)} \tag{3.11}$$

$\mathcal{P}(A)$ and $\mathcal{P}(B)$ denote the prior probabilities of $A$ and $B$. If we identify the event $A$ with the case that model $m_i$ is the correct one and event $B$ with the case that we observe data $D$ which denotes our dependent time series $y_t$, we can derive the following expression for posterior model probability:

$$\mathcal{P}(m_i|D) = \frac{\mathcal{P}(D|m_i)\mathcal{P}(m_i)}{\mathcal{P}(D)} \tag{3.12}$$

$\mathcal{P}(m_i)$ is the prior model probability, where we can make assumptions about the model. As we do not make any assumptions beforehand, this value is $\#\mathcal{M}$ for all models. $\mathcal{P}(D)$ is the total probability $\mathcal{P}(D) = \sum_{m\in\mathcal{M}} \mathcal{P}(D|m)\mathcal{P}(m)$ which can also be treated as constant.
The following theorem, i. e. the so called BIC approximation, is essentially taken from Raftery (1995).

**Theorem 3.1.** *For large samples and under the constraint of equal prior model probabilities, the following proportionality holds:*

$$\mathcal{P}(m_i|D) \propto \mathcal{P}(D|m_i) \approx \exp\Big(-\frac{BIC}{2}\Big) \tag{3.13}$$

*The Bayesian information criterion (BIC) is defined as*

$$BIC = k \ln n - 2 \ln \hat{L},$$

*where $k$ denotes the number of parameters estimated by the model, $n$ denotes the sample size and $\hat{L} = \mathcal{P}(D|\hat{\theta}, m_i)$ denotes the maximum likelihood with $\hat{\theta}$ being the likelihood-maximizing parameter vector.*

*Proof.* Since $\mathcal{P}(m_i)/\mathcal{P}(D)$ is the same constant for every model $m_i \in \mathcal{M}$, $\mathcal{P}(m_i|D) \propto \mathcal{P}(D|m_i)$ follows directly from equation (3.13). $\mathcal{P}(D|m_i)$ is defined as

$$\mathcal{P}(D|m_i) = \int \mathcal{P}(D|\theta_i, m_i)\mathcal{P}(\theta_i|m_i)d\theta_i. \tag{3.14}$$

$\theta_i$ denotes the parameter vector of model $m_i$. In the following, we keep a specific model and do not mention it in the equations in order to keep the complexity low.

We use the interior of the integral to define $g(\theta) := \ln(\mathcal{P}(D|\theta)\mathcal{P}(\theta))$ and consider a Taylor series expansion

$$g(\theta) = g(\overline{\theta}) + (\theta - \overline{\theta})^T g'(\overline{\theta}) + \frac{1}{2}(\theta - \overline{\theta})^T g''(\overline{\theta})(\theta - \overline{\theta}) + o(||\theta - \overline{\theta}||^2),$$

$\overline{\theta}$ being the value of $\theta$ that maximizes $g$. $g'(\theta)$ denotes the vector of first partial derivatives and $g''(\theta)$ is the Hessian matrix of second partial derivatives. $g'(\overline{\theta}) = 0$ because $\overline{\theta}$ maximizes $g$. Thus

$$g(\theta) \approx g(\overline{\theta}) + \frac{1}{2}(\theta - \overline{\theta})^T g''(\overline{\theta})(\theta - \overline{\theta}).$$

This estimation is good for large $n$, since the $\theta$ with a large distance to $\overline{\theta}$ have a low probability and therefore contribute little to the integral (3.14). Thus

$$\begin{aligned}
\mathcal{P}(D) &= \int \mathcal{P}(D|\theta)\mathcal{P}(\theta)d\theta \\
&= \int \exp\big(g(\theta)\big)d\theta \\
&\approx \exp\big(g(\overline{\theta})\big) \int \exp\Big(\frac{1}{2}(\theta - \overline{\theta})^T g''(\overline{\theta})(\theta - \overline{\theta})\Big)d\theta \\
&= \exp\big(g(\overline{\theta})\big)(2\pi)^{\frac{k}{2}}|A|^{-\frac{1}{2}}.
\end{aligned} \tag{3.15}$$

In the last step, we used that the density of the multivariate normal distribution integrates to one, with $A := -g''(\overline{\theta})$ and $|A|$ denoting the determinant of $A$.

$$\begin{aligned}
\int \exp\Big(\frac{1}{2}(\theta - \overline{\theta})^T g''(\overline{\theta})(\theta - \overline{\theta})\Big)d\theta &= \int \exp\Big(-\frac{1}{2}(\theta - \overline{\theta})^T A(\theta - \overline{\theta})\Big)d\theta \\
&= \sqrt{(2\pi)^k |A^{-1}|} \\
&= (2\pi)^{\frac{k}{2}}|A|^{-\frac{1}{2}}
\end{aligned}$$

Since the error in equation (3.15) is $\mathcal{O}(n^{-1})$ according to Tierney and Kadane (1986) we get

$$\ln \mathcal{P}(D) = \ln \mathcal{P}(D|\overline{\theta}) + \ln \mathcal{P}(\overline{\theta}) + \frac{k}{2}\ln(2\pi) - \frac{1}{2}\ln|A| + \mathcal{O}(n^{-1}).$$

It is known that the inverse variance-covariance matrix $A$ converges to the Fisher information matrix in large samples. According to Raftery (1995) $|A| \approx n^d|\mathbf{i}|$ with an $\mathcal{O}(n^{-1/2})$ error term. Thus,

$$\ln \mathcal{P}(D) = \ln \mathcal{P}(D|\overline{\theta}) + \ln \mathcal{P}(\overline{\theta}) + \frac{k}{2}\ln(2\pi) - \frac{k}{2}\ln n - \frac{1}{2}\ln|\mathbf{i}| + \mathcal{O}(n^{-\frac{1}{2}}). \tag{3.16}$$

Most of the terms in equation (3.16) are of order $\mathcal{O}(1)$ or less. Finally, we arrive at

$$\ln \mathcal{P}(D) = \ln \mathcal{P}(D|\overline{\theta}) - \frac{k}{2}\ln n + \mathcal{O}(1).$$

By plugging in the definition of the BIC, we can prove the claimed approximation:

$$\exp\Big(-\frac{BIC}{2}\Big) = \exp\Big(\ln(\hat{L}) - \frac{k}{2}\ln n\Big) \approx \exp\Big(\ln \mathcal{P}(D|\overline{\theta}) - \frac{k}{2}\ln n + \mathcal{O}(1)\Big) = \mathcal{P}(D)$$

□

The error term of order $\mathcal{O}(1)$ shows that the error in this estimate does not disappear even with an infinite amount of data. Raftery (1995) discuss this further and show that the error converges to zero with the correct choice of prior distribution. We will not go into this further as the BIC approximation has already been proved at this point.
For our modified version of the BMA method, instead of using this BIC approximation we follow Burnham and Anderson (2002) and base our model weights in (3.3) on the slightly different Akaike information criterion which is defined as

$$AIC = -2\ln(\hat{L}) + 2k.$$

### 3.4 Elastic Net

In this section, we will first explain how OLS, Lasso, and Ridge regression differ from each other. We will then describe the Elastic Net, which was introduced by Zou and Hastie (2005) and combines these shrinkage methods.
The basis for all methods is an OLS regression (ordinary least squares) as described in (3.1). We will now use a simpler notation:

$$y = \mathbf{X}\beta + \varepsilon$$

The design matrix $\mathbf{X} \in \mathbb{R}^{T\times(20\cdot(L+1)+K+1)}$ contains the values of all 20 exogenous variables and can also contain $L$ lags of these variables and $K$ lags of the target variable. $\beta \in \mathbb{R}^p$, $p = 20 \cdot (L+1) + K + 1$, contains all corresponding regression coefficients and the intercept. $\hat{\beta}_{OLS} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'y$ is the solution of the optimization problem

$$\min_{\beta} \sum_{t=1}^{T} (y_t - \mathbf{X}_t\beta)^2.$$

*Ridge* regression was developed to circumvent the problem of collinearity between multiple explanatory variables. McDonald and Schwing (1973) were able to show that this approach can be helpful in some cases-unlike other methods that work by eliminating variables. *Ridge* regression is described in detail in McDonald (2009). We will only briefly describe the model here. The minimization problem is extended by a penalty term that penalizes high absolute values of beta more severely. This depends on a parameter $\lambda \in [0, \infty)$.

$$\hat{\beta}_{Ridge}(\lambda_1) := \arg\min_{\beta} \left[ \sum_{t=1}^{T} (y_t - \mathbf{X}_t\beta)^2 + \lambda_1 \sum_{j=1}^{p} \beta_i^2 \right] = (\mathbf{X}'\mathbf{X} + \lambda_1\mathbf{I})^{-1}\mathbf{X}'y$$

*Lasso* regression (least absolute shrinkage and selection operator) was developed by Tibshirani (1996) and combines two different approaches to compensate for the weaknesses of OLS regression. Similar to *Ridge* regression, a *Lasso* model can shrink parameters and thus increase the prediction accuracy of the model. Unlike *Ridge*, *Lasso* can also set these parameters to zero and thus perform variable selection. This puts the focus on the more important features and makes the results easier to interpret. This is done by the following minimization problem:

$$\hat{\beta}_{Lasso} := \arg\min_{\beta} \sum_{t=1}^{T} (y_t - \mathbf{X}_t\beta)^2, \quad \text{subject to} \quad \sum_{j} |\beta_j| < \lambda_2,\ \lambda_2 \geq 0$$

Since $||\beta||_1$ is not differentiable, the solution set is angular, with corners on the axes. The optimum is often located at one of these corners, where one or more coefficients are equal to zero.
Zou and Hastie (2005) describes two specific weaknesses of the *Lasso* method. On the one hand, if there are more regressors than observations, no more regressors can be taken into account than there are observations. Second, there are cases with few regressors, many observations, and high correlation between some regressors where *Ridge* regression is superior. Zou and Hastie (2005) therefore developed the elastic net, which solves the problems of *Lasso* and works similarly in

cases where *Lasso* performs well. Here, an $L_1$ and an $L_2$ penalty term are added, which are weighted with the parameters $\lambda_1, \lambda_2 \in [0, \infty)$:

$$\hat{\beta} = \arg\min_{\beta} \left[ \sum_{t=1}^{T} (y_t - \mathbf{X}_t \beta)^2 + \lambda_1 \sum_{j=1}^{p} \beta_j^2 + \lambda_2 \sum_{j=1}^{p} |\beta_j| \right]$$

With $\alpha := \lambda_1/(\lambda_1 + \lambda_2)$ we have to solve the optimization problem

$$\arg\min_{\beta} \left[ \sum_{t=1}^{T} (y_t - \mathbf{X}_t \beta)^2 \right] \text{ subject to } (1-\alpha)||\beta||_1 + \alpha||\beta||_2 \leq t \text{ for some } t.$$

For $\alpha = 0$, we obtain the convex, but not strictly convex, *Lasso* penalty. For $\alpha = 1$, this becomes *Ridge* regression. By choosing alpha in $(0, 1)$, we obtain a strictly convex but not differentiable penalty function. This allows us to combine the positive shrinkage properties of *Ridge* and the variable selection of *Lasso* in one model. However, since this leads to overshrinkage, the estimator must be corrected by a factor:

$$\hat{\beta}_{elastic\,net} = (1 + \lambda_1)\hat{\beta}$$

Another advantage of Elastic Net over *Lasso* is the grouping effect. This refers to the property that the model assigns very similar coefficients to highly correlated variables. With a group of highly correlated or even identical variables, *Lasso* would select one of them to remain in the model. Elastic Net, on the other hand, recognizes this group of variables and assigns them similar coefficients accordingly.

For implementation, we use the function **glmnet** of the R package **GLMNET**. We perform a grid search with zero and one exogenous lag and $\alpha \in \{0, 0.2, 0.4, 0.6, 0.8, 1\}$. In three sectors the *Ridge* regression turned out to be the dominant method, the *Lasso* estimation is superior in only one economic sector. We will refer to the Elastic Net estimation as *ELN*. Note that $\alpha$ is defined differently in the R package and *Ridge* regression is performed with $\alpha = 0$.

The following hyperparameters turned out to be the best:

| Para | Financial | NFC-nonRE | Sov | NFC-RE | HH-nonRE | HH-RE | Sov-HR |
|---|---|---|---|---|---|---|---|
| $\alpha$ | 0.4 | 1.0 | 0.0 | 0.4 | 0.2 | 0.0 | 0.0 |
| $L$ | 1 | 0 | 1 | 0 | 0 | 1 | 1 |

## 3.5 Tree Based Models

Let us first explain how a binary regression tree works and then how we can combine multiple trees to get a sum-of-trees model. A regression tree is a non-parametric model which helps us to make predictions about a target variable based on a vector of explanatory variables. Let $x_t$ be the vector of explanatory variables – a row of $\mathbb{X}$ from Definition (2.2) at time $t$ and $x_t^j$ the $j$-th component of $x_t$. Let $\mathcal{X}$ denote the *feature space*, the space which all the vectors of the explanatory variables are elements of, i. e. $\mathcal{X} = \mathbb{R}^d$, $d$ being the number of explanatory variables. The core idea of regression trees is to divide $\mathcal{X}$ into disjoint regions $R_1, R_2, \ldots, R_B \subset \mathcal{X}$ with $R_1 \cup R_2 \cup \cdots \cup R_B = \mathcal{X}$. The prediction for $y$-values will be the mean of the observations in each region.
To perform a single split we have to define a subset $A \subset \mathcal{X}$ such that either $x_t \in A$ or $x_t \notin A$. Typically, these decision rules are based on a single component $x_t^j \in \mathbb{R}$ of $x_t$ and

$$A = \{x \in \mathcal{X} | x_j \leq c\}, \quad c \in \mathbb{R}. \tag{3.17}$$

For a given region $A$ (or $A^C = \mathcal{X} \setminus A$), the prediction for the $y$-value belonging to $x \in A$ (or $x \in A^C$) is the mean of $y$ in that region:

$$\hat{y}_A = \frac{1}{|A|} \sum_{t=1}^{T} y_t \cdot \mathbf{1}_{\{x_t \in A\}}, \quad \hat{y}_{A^C} = \frac{1}{|A^C|} \sum_{t=1}^{T} y_t \cdot \mathbf{1}_{\{x_t \in A^C\}}$$

The parameter $c \in \mathbb{R}$ and the feature $j$ is selected so that the sum of squared errors

$$SSE(j, c) = \sum_{t=1}^{T} \left(y_t - \hat{y}_A\right)^2 \cdot \mathbf{1}_{\{x_t \in A\}} + \left(y_t - \hat{y}_{A^C}\right)^2 \cdot \mathbf{1}_{\{x_t \in A^C\}}$$

is minimized. Once the optimal split of $\mathcal{X}$ is found, this process is repeated recursively for the regions $A$ and $A^C$ until a stopping criterion is met or no further reduction in the SSE can be achieved. We receive the $B$ disjoint regions of $\mathcal{X}$ mentioned above and a set of parameter values $M = \{\mu_1, \ldots, \mu_B\}$. Each parameter value in $M$ belongs to one of the regions and denotes the predicted target value in that region:

$$\mu_i := \hat{y}_{R_i} = \frac{1}{|R_i|} \sum_{t=1}^{T} y_t \cdot \mathbf{1}_{\{x_t \in R_i\}}$$

**Definition 3.1.** Let $T$ denote a binary regression tree consisting of a set of interior node decision rules of the form (3.17) and a set of terminal nodes. Let $M = \{\mu_1, \ldots, \mu_B\}$ denote a set of parameter values associated with each of the $B$ terminal nodes of $T$. A *single-tree model* is defined by the equation

$$y = g(x; T, M) + \varepsilon, \qquad \varepsilon \sim \mathcal{N}(0, \sigma^2), \tag{3.18}$$

where $g(x; T, M)$ is the function that assigns $\mu_i$ to $x$ if $x \in R_i$.

Now we consider $m$ binary trees, $m \in \mathbb{N}$, each of which has different decision rules and a different

set of parameter values. The number of terminal nodes per tree can also vary.

**Definition 3.2.** Let $T_j$, $j = 1, \ldots, m$ be binary trees and let $M_j$ be the set of parameter values associated with $T_j$. A *sum-of-trees model* is defined by the model equation

$$y = \sum_{j=1}^{m} g(x; T_j, M_j) + \varepsilon, \qquad \varepsilon \sim \mathcal{N}(0, \sigma^2). \tag{3.19}$$

Since the model can also contain trees that depend on a single feature, we can also model the main effects of the features. Trees that depend on two or more features, i. e. explanatory variables, can represent different interaction effects of different orders. Thus, with a large number of trees $m$, the model offers a high degree of representational flexibility.

### 3.5.1 Bayesian Additive Regression Trees

*Bayesian Additive Regression Trees (BART)* were introduced by Chipman et al. (2010) and represent a statistical learning method that combines machine learning with a Bayesian framework. *BART* models the response variable as a sum of regression trees, each constrained by a regularization prior, which allows capturing non-linear interactions within the data. Guth (2022) compared 43 methods in the context of credit risk stress testing to predict default probabilities. *BART* turned out to be the overall winner method. It is also widely used to analyze causal inference and treatment effects, e. g. by Dorie et al. (2022). For implementation, we use the function **wbart** of the R package **BART**.

For fixed $m$, a sum-of-trees model is determined by $(T_1, M_1), \ldots, (T_m, M_m)$ and $\sigma$. Large trees and a large $m$ can usually lead to redundancy across the tree components in the sense that many different choices of the trees result in the same or a very similar model. Next, for a fully Bayesian approach, it is necessary to impose prior distributions on every parameter in the sum-of-trees model. The priors help us to regularize the influence of large tree components. The few small trees should retain a certain importance in representing main effects and low-order interaction effects.
Chipman et al. (2010) restrict their choice of priors to those that satisfy the following condition:

$$\begin{aligned}
\mathcal{P}\big((T_1, M_1), \ldots, (T_m, M_m), \sigma\big) &= \Big(\prod_{j=1}^{m} \mathcal{P}(T_j, M_j)\Big)\mathcal{P}(\sigma) \\
&= \Big(\prod_{j=1}^{m} \mathcal{P}(M_j|T_j)\mathcal{P}(T_j)\Big)\mathcal{P}(\sigma) \\
&= \Big(\prod_{j=1}^{m} \Big(\prod_{i=1}^{B_j} \mathcal{P}(\mu_{ij}|T_j)\Big)\mathcal{P}(T_j)\Big)\mathcal{P}(\sigma)
\end{aligned}$$

$B_j$ denotes the number of terminal leaves of tree $T_j$, i. e. $\#M_j$, $\mu_{ij} \in M_j$. They use the same forms for all $\mathcal{P}(T_j)$ and for all $\mathcal{P}(\mu_{ij}|T_j)$. Only three different priors need to be developed in this way.

**Prior for $\mathcal{P}(T_j)$**

To determine this prior Chipman et al. (2010) follow the recommendations by Chipman et al. (1998). There are three aspects that specify $\mathcal{P}(T_j)$, namely:

1) The splitting probability of a node

2) The distribution on the splitting variable assignment

3) The distribution on the splitting rule assignment

Let $d_\eta \in \mathbb{N}_0$ denote the depth of node $\eta$, i. e. the number of splits above $\eta$. The probability of node $\eta$ being split is

$$\mathcal{P}_{\text{split}}(\eta) = \alpha(1 + d_\eta)^{-\beta}, \qquad \alpha \in (0, 1),\ \beta \in \mathbb{R}_0^+.$$

$\alpha$ is the probability that the first node of the tree will be split, $\beta$ is a parameter to control for the shape of the trees. A higher $\beta$ makes deeper nodes less likely to split, which puts higher probability on "bushy" trees whose terminal nodes do not vary much in depth. Next, we impose a uniform distribution on the variables that can be used to split the feature space. This means that when a node is split, each explanatory variable has the same probability of being the feature on which the decision rule is based. This choice represents the prior information that at each node available predictors are equally likely to be effective. Finally, we choose the parameter $c$ uniformly from the available observed values of the previously chosen predictor. This choice represents the prior information that for each available predictor the available split values are equally likely to be effective.
Chipman et al. (2010) recommend keeping the individual trees quite small by using the parameters $\alpha = 0.95$ and $\beta = 2$. In this case 92% of the trees in the sum-of-trees model will have between two and four terminal nodes.

**Prior for $\mathcal{P}(\mu_{ij}|T_j)$**

In the sum-of-trees model, the following equation for the conditional expectation value of $y$ holds:

$$\mathbb{E}[y|x] = \sum_{j=1}^{m} \mu_{i_j j}, \qquad 1 \leq i_j \leq B_j$$

Here, $i_j$ is the index of the leaf of tree $j$ to which $x$ belongs. We impose a normal distribution with parameters $\mu_\mu$ and $\sigma_\mu^2$ on the $\mu_{ij}$'s. Since they are a priori independent and identically distributed, $\mathbb{E}[y|x]$ has the a priori distribution $\mathcal{N}(m\mu_\mu, m\sigma_\mu^2)$.
The aim now is to choose the parameters of the distribution so that it produces plausible values for $y$ with a high probability. The minimum and maximum values $y_{min}$ and $y_{max}$ of the observations $y$ in the data set are used as a reference. By setting

$$m\mu_\mu - k\sqrt{m}\sigma_\mu = y_{min} \quad \text{and} \quad m\mu_\mu + k\sqrt{m}\sigma_\mu = y_{max}$$

for a preselected parameter $k$ we can control the highly probable region of our prior.

Now we shift and rescale the time series of the endogenous variable $y$ such that $y_{min} = -0.5$ and $y_{max} = 0.5$. Now the parameters are

$$\mu_\mu = 0 \quad \text{and} \quad \sigma_\mu = \frac{1}{2k\sqrt{m}}.$$

Using $\mu_{ij} \sim \mathcal{N}(0, \sigma_\mu^2)$ we shrink the tree parameters towards zero. This allows us to reduce the influence of a single tree on the whole model. According to Chipman et al. (2010), $k$ between 1 and 3 gives good results, $k = 2$ is recommended as a default. Since the structure of the regression trees is invariant to monotonic transformations of the explanatory variables, we can simply transform $y$ without taking any further steps.

**Prior for $\mathcal{P}(\sigma)$**
As the prior distribution of $\sigma^2$ we use an inverse chi-square distribution $\sigma^2 \sim \nu\lambda/\chi^2_\nu$, $\lambda$ being a scale parameter, and $\nu$ denoting the number of degrees of freedom. Again, we use the given data to obtain a plausible form of the distribution of $\sigma$. As an estimation $\hat{\sigma}$ of $\sigma$ we can use either the standard deviation of $y$ or the residual standard deviation of a least squares regression of $y$ on all explanatory variables. First we pick a value for $\nu$ and then choose $\lambda$ so that

$$\mathcal{P}(\sigma < \hat{\sigma}) = q$$

for a chosen percentile parameter $q \in (0, 1)$. Chipman et al. (2010) recommend the default setting $(\nu, q) = (3, 0.9)$. A higher $q$ or less degrees of freedom would lead to a more aggressive setting in which much mass is concentrated in a relatively small interval.

**Sampling the posterior distribution**
Our next goal is to derive the posterior distribution

$$\mathcal{P}((T_1, M_1), \ldots, (T_m, M_m), \sigma | y), \tag{3.20}$$

given the observed data $y$. For this purpose, Chipman et al. (2010) use a Gibbs sampler. This means that $m$ pairs $(T_j, M_j)$, $j = 1, \ldots, m$ are drawn successively, while all other trees $(T_k, M_k)$, $k \neq j$ remain unchanged. The draw $(T_j, M_j)$ therefore only depends on the residual

$$R_j = y - \sum_{k \neq j} g(x; T_k, M_k).$$

The draw from $(T_j, M_j)|R_j, \sigma$ is done in two successive steps. First, we draw the tree structure from $T_j|R_j, \sigma$ using the Metropolis-Hastings algorithm described in Chipman et al. (1998). This algorithm uses the $T_j$ from the previous iteration and either grows a node by splitting a terminal node into two new ones (GROW), turns a parent of two terminal nodes into a terminal node by collapsing its children (PRUNE), changes the splitting rule of an internal node (CHANGE), or swaps the splitting rules of a parent-child pair that are both internal nodes (SWAP). The action and the node on which it is performed are selected at random. Second, we draw the set of parameter values $M_j|T_j, R_j, \sigma$ from a normal distribution for each $\mu_{ij}$. Having done this, we

are ready to calculate the subsequent residual $R_{j+1}$. After the $m$ draws of $(T_j, M_j)$ we draw $\sigma|T_1, \ldots, T_m, M_1, \ldots, M_m, y$ from the previously described inverse gamma distribution. This hole process represents one iteration of our algorithm.
The chain must be initialized with $m$ single-node trees. Then iterations are performed until a satisfactory convergence is achieved. After a sufficient burn-in period, we continue by generating a sequence of $(T_1, M_1), \ldots, (T_m, M_m), \sigma$ of length $K$ that, according to Chipman et al. (2010), is converging in distribution to the posterior (3.20). To estimate the $y$ value for a given $x$ we can approximate the posterior mean $\mathbb{E}[f(x)|y]$ by calculating

$$\frac{1}{K}\sum_{k=1}^{K} f^k(x), \qquad f^k(x) := \sum_{j=1}^{m} g(x; T_j^k, M_j^k),$$

where $(T_j^k, M_j^k)$ denotes $(T_j, M_j)$ from the $k$-th place in our generated sequence.

**Algorithm 3** Posterior Sampling

Initialize $T_1^0, \ldots, T_m^0, M_1^0, \ldots, M_m^0$ as single-node trees
**while** No satisfactory convergence in distribution **do**
    Metropolis-Hastings algorithm for each $j \in \{1, \ldots, m\}$
    Sample $\sigma$ from inverse gamma distribution
**end while**
Overwrite $T_1^0, \ldots, T_m^0, M_1^0, \ldots, M_m^0$ with last trees from previous loop
**for** $k = 1$ to $K$ **do**
    **for** $j = 1$ to $m$ **do**
        Build $T_j^k$ using MH algorithm on $T_j^{k-1}$
        Build $M_j^k$ by sampling from normal distribution
        Sample new $\sigma$
    **end for**
**end for**

We analyze the following hyperparameters:

| Para | Financial | NFC-nonRE | Sov | NFC-RE | HH-nonRE | HH-RE | Sov-HR |
|---|---|---|---|---|---|---|---|
| $m$ | 50 | 200 | 200 | 200 | 200 | 200 | 200 |
| $k$ | 2.5 | 2.5 | 2 | 2.5 | 2.5 | 2.5 | 2.5 |
| $\nu$ | 5 | 7 | 7 | 5 | 7 | 5 | 3 |
| $q$ | 0.9 | 0.75 | 0.9 | 0.75 | 0.9 | 0.75 | 0.99 |
| $L$ | 0 | 1 | 0 | 0 | 1 | 1 | 0 |

### 3.5.2 Extreme Gradient Boosting

*Extreme Gradient Tree Boosting (XGBoost)* is a machine learning system for tree boosting developed by Chen and Guestrin (2016). A single regression tree acts as a weak learner here. For each iteration, a new tree is added to the existing model that best improves the model. The stronger ones among the weak learners are weighted higher, whereas the really weak ones are ignored. According to Chen and Guestrin (2016), *XGBoost* has been superior to other methods in many challenges in machine learning and data mining. For implementation, we use the func-

tion **xgb.train** of the R package **xgboost**.

A data point is given by $(x_t, y_t)$, $x_t \in \mathbb{R}^d$, $y_t \in \mathbb{R}$, $t \in \{1, \dots, T\}$. We use equation (3.19) to define the function $\phi$ that estimates the target value for a single data point:

$$\hat{y}_t = \phi(x_t) := \sum_{j=1}^{m} g(x_t, T_j, M_j)$$

In each iteration step we want to minimize

$$\mathcal{L}(\phi) := \sum_{t=1}^{T} l(\hat{y}_t, y_t) + \sum_{j=1}^{m} \Omega(T_j.M_j),$$
$$\Omega(T_j, M_j) := \gamma B_j + \frac{1}{2}\lambda ||\mu^{(j)}||^2, \quad \mu^{(j)} = (\mu_{1j}, \dots, \mu_{B_j j}),$$

where $l$ is a differentiable convex loss function that measures the quality of the prediction and $\Omega$ penalizes the complexity of the trees used in the model. Penalizing complex regression trees helps to avoid large trees and over-fitting. In each iteration, we add the tree which most improves the current model. With $\hat{y}_t^{(i)}$ being the prediction for $y_t$ after the $i$-th iteration, we want to add the tree $(T_i, M_i)$ which minimizes

$$\begin{aligned}
\mathcal{L}^{(i)} &:= \sum_{t=1}^{T} l(y_t, \hat{y}_t^{(i-1)} + g(x_t, T_i, M_i)) + \Omega(T_i, M_i) \\
&\approx \sum_{t=1}^{T} \big[ l(y_t, \hat{y}_t^{(i-1)}) + \frac{\partial l(y_t, \hat{y}_t^{(i-1)})}{\partial \hat{y}_t^{(i-1)}} g(x_t, T_i, M_i) + \frac{1}{2} \frac{\partial^2 l(y_t, \hat{y}_t^{(i-1)})}{\partial (\hat{y}_t^{(i-1)})^2} (g(x_t, T_i, M_i))^2 \big] + \Omega(T_i, M_i).
\end{aligned}$$

We can simplify this term by omitting the constant part and restructuring the sum. Let $I_b = \{t | x_t \in R_b\}$ be the instance set of leaf $b$. Then our objective looks as follows:

$$\begin{aligned}
\overline{\mathcal{L}}^{(i)} &= \sum_{t=1}^{T} \big[ \frac{\partial l(y_t, \hat{y}_t^{(i-1)})}{\partial \hat{y}_t^{(i-1)}} g(x_t, T_i, M_i) + \frac{1}{2} \frac{\partial^2 l(y_t, \hat{y}_t^{(i-1)})}{\partial (\hat{y}_t^{(i-1)})^2} (g(x_t, T_i, M_i))^2 \big] + \Omega(T_i, M_i) \\
&= \sum_{t=1}^{T} \big[ \frac{\partial l(y_t, \hat{y}_t^{(i-1)})}{\partial \hat{y}_t^{(i-1)}} g(x_t, T_i, M_i) + \frac{1}{2} \frac{\partial^2 l(y_t, \hat{y}_t^{(i-1)})}{\partial (\hat{y}_t^{(i-1)})^2} (g(x_t, T_i, M_i))^2 \big] + \gamma B_i + \frac{1}{2}\lambda \sum_{b=1}^{B_i} \mu_{bi}^2 \\
&= \sum_{b=1}^{B_i} \left[ \Big( \sum_{t \in I_b} \frac{\partial l(y_t, \hat{y}_t^{(i-1)})}{\partial \hat{y}_t^{(i-1)}} \Big) \mu_{bi} + \frac{1}{2} \Big( \sum_{t \in I_b} \frac{\partial^2 l(y_t, \hat{y}_t^{(i-1)})}{\partial (\hat{y}_t^{(i-1)})^2} + \lambda \Big) \mu_{bi}^2 \right] + \gamma B_i
\end{aligned}$$

Once we have fixed a tree structure $T_i$ for the $i$-th iteration, we differentiate $\overline{\mathcal{L}}^{(i)}$ by $\mu_{bi}$ and calculate the optimal weights $M_i$ by

$$\mu_{bi}^* = -\frac{\sum_{t \in I_b} \frac{\partial l(y_t, \hat{y}_t^{(i-1)})}{\partial \hat{y}_t^{(i-1)}}}{\sum_{t \in I_b} \frac{\partial^2 l(y_t, \hat{y}_t^{(i-1)})}{\partial (\hat{y}_t^{(i-1)})^2} + \lambda}.$$

To develop the structure of the tree $T_i$, the split candidates are compared step by step and

the one that best complements the previous model is selected in the sense that $\mathcal{L}$ is minimized. Split candidates are computed for each explanatory variable. As the computational complexity is usually too high to compare all candidates of a continuous variable, Chen and Guestrin (2016) use weighted quantile sketches. In this method, percentiles of the explanatory variables are used as split candidates.
With $h_t := \frac{\partial^2 l(y_t, \hat{y}_t^{(i-1)})}{\partial (\hat{y}_t^{(i-1)})^2}$ let $\mathcal{D}_k = \{(x_1^k, h_1), \dots, (x_T^k, h_T)\}$ be the multi-set of the $k$-th feature values and the corresponding second-order gradient. We define the rank function $r_k : \mathbb{R} \longrightarrow \mathbb{R}_0^+$ as

$$r_k(z) = \frac{1}{\sum_{(x,h)\in\mathcal{D}_k} h} \sum_{(x,h)\in\mathcal{D}_k, x<z} h.$$

Now we try to find candidate split points $\{s_1^k, \dots, s_l^k\} \subset \{x_1^k, \dots, x_T^k\}$ such that

$$|r_k(s_j^k) - r_k(s_{j+1}^k)| < \varepsilon, \quad s_1^k = \min_t x_t^k, \quad s_l^k = \max_t x_t^k,$$

for a given parameter $\varepsilon \in (0, 1)$. This means, there are approximately $1/\varepsilon$ split candidates per explanatory variable. After finding the best split, we repeat this procedure in the new leafs until a maximum depth of $T_i$ is reached. This is the local variant, where new candidates are calculated after each split. Although this is more computationally intensive, it is also more accurate than the global variant, in which all split candidates are only calculated once at the beginning.
We analyze the following hyperparameters:

| Para | Financial | NFC-nonRE | Sov | NFC-RE | HH-nonRE | HH-RE | Sov-HR |
|---|---|---|---|---|---|---|---|
| $\eta$ | 0.1 | 0.1 | 0.1 | 0.5 | 0.1 | 0.1 | 0.1 |
| $d$ | 8 | 6 | 8 | 8 | 4 | 8 | 4 |
| $L$ | 0 | 0 | 0 | 1 | 0 | 0 | 0 |

| Symbol | Meaning | Possible Values |
|---|---|---|
| $\eta$ | Learning Rate | 0.1, 0.3, 0.5 |
| $d$ | Maximum depth | 4, 6, 8 |
| $L$ | Exogenous Lags | 0, 1 |
| $o$ | Occam's window | 10, 20, 30 |
| $\gamma$ | Correlation coefficient | 0.7, 0.8, 0.9 |
| DW | DW-Test Statistic | 0.1, 0.2 |
| $m$ | Number of Trees | 50, 200 |
| $k$ | Prior for $\sigma_\mu$ (BART) | 1.5, 2.0, 2.5 |
| $\nu$ | Degrees of Freedom | 3, 5, 7 |
| $q$ | Quantile (BART) | 0.75, 0.9, 0.99 |
| $\alpha$ | Mixing parameter (EN) | 0, 0.2, 0.4, 0.6, 0.8, 1 |

Table 1: Declaration of Hyperparameters

# 4 Forecast Results

In this chapter, we will first explain which loss functions are suitable for analyzing the predictions. Then, we explain how to give more weight to predictions within adverse scenarios. Next, we evaluate and compare the models' estimations regarding different criteria. To maintain clarity in this subsection, we provide a detailed explanation of the results for one sector and briefly mention notable findings from the others. All figures and tables can be found in the appendix.

## 4.1 Consistent Scoring Functions

Varian (1975) compared three estimation methods in a simulation study. One method used a statistically sound and meaningful estimate. The other two estimated a constant value for the entire time series. When evaluating the mean scores of the estimates, different results were obtained depending on the scoring function. This shows that the correct choice of scoring function is of great importance. First, we will explain a few basic elements to understand which scoring functions are suitable for evaluating our predictions and start with some definitions in a decision-theoretic setting. All of the following definitions and theorems are summarized by Gneiting (2011).

**Definition 4.1.** 1) An *observation domain* $O \subseteq \mathbb{R}^d$ comprises the potential outcomes of a future observation, an *action domain* $A \subseteq \mathbb{R}^d$ comprises potential actions of a decision maker, i. e. the prediction. In our case of point forecasting $D := O = A = \mathbb{R}$. A *prediction-observation domain* is the Cartesian product $\mathcal{D} = D \times D$.

2) A *loss function* $L : A \times O \to [0, \infty)$, $L(a, o)$ represents the loss incurred when the decision maker takes the action $a \in A$ and the observation $o \in O$ materializes.

3) A *scoring function* $S : \mathcal{D} \to [0, \infty)$, $S(x, y)$ represents the loss or penalty when the point forecast $x \in D$ is issued and the observation $y \in D$ materializes.

4) Let $\mathcal{F}$ be a family of potential probability distributions for the future observation $Y$ that takes values in $D$. The *optimal point forecast* under $F \in \mathcal{F}$ for $Y$ is the Bayes rule

$$\hat{x} = \arg\min_{x \in D} \mathbb{E}_F[S(x, Y)].$$

**Definition 4.2.** A *statistical functional*, or simply a *functional*, is a mapping from a class of probability distributions to a Euclidean space, a subset of $\mathbb{R}$ in our case:

$$T : \mathcal{F} \to \mathcal{P}(D), \qquad F \mapsto T(F) \subseteq D.$$

The *mean functional* maps a probability distribution to its expectation value and is therefore single valued:

$$T_M : \mathcal{F} \to \mathbb{R}, \qquad F \mapsto T(F) = \mathbb{E}_F[Y], \qquad \text{for } Y \sim F$$

We only have to consider the mean functional $T_M$ because the regression-based models and the tree-based models estimate the mean. They first estimate the expectation value as the most

likely materialization of $y$ for different regressions/trees and then build an average of those values.

**Definition 4.3.** The scoring function $S$ is *consistent* for the functional $T$ relative to the class $\mathcal{F}$ if

$$\mathbb{E}_F[S(t,Y)] \leq \mathbb{E}_F[S(x,Y)],$$

for all $F \in \mathcal{F}$, all $t \in T(F)$ and all $x \in D$. It is *strictly consistent* if it is consistent and equality implies that $x \in T(F)$.

In other words, the expected score should be the lowest if we estimate the most likely value. In our case $T(F) = \mathbb{E}_F$ is single valued. Therefore, $\mathbb{E}_F[S(x,Y)]$ should have its minimum at $x = T(F)$.

**Definition 4.4.** The functional $T$ is *elicitable* relative to the class $\mathcal{F}$ if there exists a scoring function $S$ that is strictly consistent for $T$ relative to $\mathcal{F}$.

The following theorem is needed later when we prove our scoring functions to be consistent.

**Theorem 4.1.** *Let $T$ be elicitable relative to $\mathcal{F}$. Consider a weight function*

$$w : D \to [0, \infty).$$

*Let $\mathcal{F}^{(w)} \subseteq \mathcal{F}$ denote the subclass of probability distributions in $\mathcal{F}$ such that $F^{(w)}$ has a density proportional to $w(y)f(y)$, $f(y)$ is proportional to the density of a $F \in \mathcal{F}$, and $w(y)f(y)$ has finite integral over $D$. Define the functional*

$$T^{(w)} : \mathcal{F}^{(w)} \to \mathcal{P}(D), \qquad F \mapsto T^{(w)}(F) = T(F^{(w)}).$$

*Then the following holds:*

*1) $T^{(w)}$ is elicitable.*

*2) If $S$ is consistent for $T$ relative to $\mathcal{F}$, then $S^{(w)}(x,y) = w(y)S(x,y)$ is consistent for $T^{(w)}$ relative to $\mathcal{F}^{(w)}$.*

*3) If $S$ is strictly consistent for $T$ relative to $\mathcal{F}$, then $S^{(w)}$ is strictly consistent for $T^{(w)}$ relative to $\mathcal{F}^{(w)}$.*

**Theorem 4.2.** *Let $\mathcal{F}$ be the class of compactly supported probability measures on the interval $I \subseteq \mathbb{R}$ with finite first moment. Let $S$ be a scoring function that satisfies the follwing assumptions on the domain $\mathcal{D} = I \times I$:*

*A1) $S(x,y) \geq 0$ with equality if $x = y$*

*A2) $S(x,y)$ is continuous in $x$*

*A3) $\partial_x S(x,y)$ exists and is continuous in $x$ whenever $x \neq y$*

*Then the following holds:*

*1) The mean functional is elicitable relative to the class $\mathcal{F}$.*

*2) $S$ is consistent for the mean functional relative to $\mathcal{F}$ if, and only if, it is of the form*

$$S(x, y) = \phi(y) - \phi(x) - \phi'(x)(y - x) \tag{4.1}$$

*where $\phi$ is a convex function with subgradient on $I$, $y$ is the realization of the real value and $x$ is the forecast estimation.*

*3) If $\phi$ is strictly convex, $S$ is strictly consistent for the mean functional relative to $\mathcal{F}$ on $I$ for which both $\mathbb{E}_F[Y]$ and $\mathbb{E}_F[\phi(Y)]$ exist and are finit.*

The form (4.1) is a Bregman divergence $D_\phi(y, x) = \phi(x) - \phi(y) - \phi(y)(x - y) = S(x, y)$. We will use the following scoring functions, some of which are consistent for the mean functional, to evaluate our estimation results.

**Squared Loss**

The first scoring function that we use is the squared loss $S_1(x, y) := (x - y)^2$. By averaging over all scores this results in the *mean squared error (MSE)*. We can easily prove that $S_1$ is consistent for the mean functional by setting $\phi(x) := x^2$. Since $\phi''(x) = 2$, $\phi$ is strictly convex. Then,

$$S(x, y) = y^2 - x^2 - 2x(y - x) = x^2 + y^2 - 2xy = (x - y)^2.$$

**Exponential Loss**

Our second scoring function is called *exponential loss* and is defined by $S_2(x, y) := e^y - e^x - e^x(y - x)$. Again, to prove its consistency, we have to show that $S_2$ is of the form (4.1). We do so by simply setting $\phi(x) := \exp(x)$ which is a strictly convex function. Consistency follows directly from Theorem 4.2.

**Linear Exponential Loss**

The linear exponential loss function, developed by Varian (1975), is a asymmetrical function defined by

$$S_3(x, y; a) := e^{a(x-y)} - a(x - y) - 1, \qquad a \in \mathbb{R} \setminus \{0\}.$$

This is a mix of the exponential and linear loss function. We can freely select the parameter a to adjust the function as desired. For $a > 0$, the score is particularly high when $x > y$. In other words, overestimates are penalized more severely than underestimates. If we select $a < 0$, the score grows exponentially with $y - x$. In this case, underestimations are penalized more heavily. For the analysis of the estimates, we choose $a = 1$ and $a = -1$:

$$S_{3+}(x, y) = S_3(x, y; 1) = e^{x-y} - (x - y) - 1 \quad \text{and}$$
$$S_{3-}(x, y) = S_3(x, y; -1) = e^{y-x} - (y - x) - 1$$

To prove the consistency of $S_3(x, y; a)$ we pick $\phi(x; a) = 1/a^2 \exp(ax)$. Since $\phi''(x; a) =$

$\exp(ax) > 0$ for all $x \in \mathbb{R}$, $\phi$ is strictly convex. It holds

$$\begin{aligned} D_{\phi(\cdot;a)}(x,y) &= \phi(x;a) - \phi(y;a) - \phi'(y;a)(x-y) \\ &= \frac{1}{a^2}e^{ax} - \frac{1}{a^2}e^{ay} - \frac{1}{a}e^{ay}(x-y) \\ &= \frac{1}{a^2}\Big[e^{ax} - e^{ay} - ae^{ay}(x-y)\Big] \\ &= \frac{1}{a^2}e^{ay}\Big[e^{a(x-y)} - a(x-y) - 1\Big] \\ &= \frac{1}{a^2}e^{ay}\, S_3(x,y;a) =: \overline{S}_3(x,y;a). \end{aligned}$$

$\overline{S}_3$ is therefore a Bregman divergence and strictly consistent for the mean functional relative to $\mathcal{F}$ by Theorem 4.2. Define the weight function

$$w(y;a) := a^2 e^{-ay}.$$

Then, by Theorem 4.1, $S_3(x,y;a) = w(y;a) \cdot \overline{S}_3(x,y;a)$ is strictly consistent for the weighted mean functional $T^{(w)}$ relative to the class of weighted probability distributions $\mathcal{F}^{(w)}$.
Although the consistency of $S_3$ is not ensured for the non-weighted mean functional, it will also be used to compare estimation results. An asymmetrical scoring function could provide new and interesting insights, and we could not find any consistent ones defined on negative real numbers.

## 4.2 Weighting by Time Period

In the context of stress testing, reliable predictions in adverse scenarios are particularly important. Adverse scenarios are characterized by extremely unusual behavior of macroeconomic variables. Since they occur very infrequently, our already short time series contain hardly enough data to test the models in adverse scenarios with sufficient accuracy. For this reason, the aim of this chapter is to present a method for giving more weight to erroneous estimates in crisis years. The *European Financial Crisis Database* forms the basis for the definition of crisis periods. It is freely available on the website of the **European Systemic Risk Board** (ESRB).[2] The global economic crisis began in August 2007 and lasted until June 2013 in Germany, according to the ESRB. In addition, another economic crisis, triggered by the outbreak of COVID-19, began in March 2020 and lasted until April 2021.
Gneiting and Ranjan (2011) state the following theorem:

**Theorem 4.3.** *Suppose that $f$ is the sampling density of the random variable $Y$. Let $S_0$ be any proper scoring rule and let $w$ be a weight function such that $0 < \int w(y)f(y)dy < \infty$. Then the expected value of the weighted score*

$$S(g, Y) = w(Y)S_0(g, Y)$$

*is minimized if we issue the density forecast*

$$g(y) = \frac{w(y)f(y)}{\int w(y)f(y)dy}.$$

This means that a weighted scoring rule is inappropriate for the original density prediction. This result also holds for consistent scoring functions, since they are a special case of proper scoring rules. If we add an observation-based weighting function to a consistent scoring function, this new scoring function will no longer be consistent for the same function. For more on scoring rules and how they differ from scoring functions, see Gneiting (2011). Taggart (2022) shows how a scoring function can be partitioned to focus on specific regions, such as the tails of the distribution of observations, without losing the consistency property. The weight depends mainly on the region in which the observation $y$ is located.
Since we cannot say with certainty how PDs in different economic sectors will behave in adverse scenarios, the weighting should not be based on observation $y$. Instead, we assign a higher weight to forecasts that fall within a crisis period, regardless of the actual value of the estimate. This means that we artificially multiply forecasts and scores in crisis periods. Assuming that the definition of crisis periods depends only on macroeconomic variables and not on the PDs themselves, the weighting is independent of observation $y$. This preserves the consistency of the scoring function.
Let $y_1, \ldots, y_t$ for $t = 4, \ldots, 42$ denote the training time series of Delta-Logit-Default Probabilities and $\widehat{y}_t^h$, $h = 1, \ldots, 12$ the forecast time series for the following twelve quarters $y_{t+h}$ for each $t$. Let $\hat{y}_t^P := (\hat{y}_t^1, \ldots, \hat{y}_t^{12})$ denote the tuple of the twelve forecasts from time point $t$. We now

[2] https://www.esrb.europa.eu/pub/financial-crises/html/index.en.html

simply multiply the occurrence of $\hat{y}_t^P$ by the number of forecasts in $\hat{y}_t^P$ that are within a risk horizon. The new quantity of the tuple $\hat{y}_t^P$ is denoted by

$$Q_t := 1 + \sum_{h=1}^{12} \mathbb{1}_{\{t+h \in T_R\}}, \qquad T_R := \{n \in \mathbb{N} \,|\, 5 \leq n \leq 16 \ \vee \ 43 \leq n \leq 47\}.$$

Since our data series starts in 2008Q1 and the training time series starts in 2009Q3 due to lags, $t = 5$ corresponds to 2010Q3 and $t = 47$ corresponds to 2021Q1.
This also means that forecasts made shortly before or after a crisis period are weighted slightly more heavily. This makes sense, as it results in a kind of smooth transition at the boundaries of the crisis periods. We will use this weighting exclusively for prediction scores in the next chapter and compare them with the unweighted version.

## 4.3 Description of Results

The initial step in the research process involves the comparison of the graphs that represent the forecasts. This is followed by an analysis of the distribution of forecast errors, with the objective of ascertaining whether there is a discrepancy in their nature depending on the distance of the predicted quartal. Here, we use four different scoring functions and point out their effects on the results. Subsequently, the Diebold-Mariano test is employed to determine whether the errors differ significantly among the analyzed methods. Finally, we analyze the feature importance of several models with respect to differences between different economic sectors and different forecasting methods.
To analyze the results, we transform the Delta-Logit-PDs from (2.2) and their forecasts back into normal PDs by calculating

$$\begin{aligned} \hat{z}_t &= z_{t-4} + \hat{y}_t, \qquad t = 5, \dots, 8 \\ \hat{z}_t &= \hat{z}_{t-4} + \hat{y}_t, \qquad t = 9, \dots, 60 \end{aligned} \tag{4.2}$$

iteratively. Afterwards, we apply the back-transformation

$$\widehat{PD} = \frac{1}{1 + \exp(-\hat{z})} \tag{4.3}$$

in order to get plausible probability values. We will use these transformed forecasts in all the following chapters.

### 4.3.1 Forecasts

Here we show only the forecasts for the financial institutes sector as an example. The graphs for the other six economic sectors can be found in the appendix.
In the top left corner of Figure 2, we can see the time series of forecasts for the next time point. The *BMA* forecasts, i. e. the forecasts for which we used predefined methods for the standard Bayesian Model Averaging, are not shown here because the PDs are so high that they disrupt the figure. Initially, with a very short training time series, the predictions for the next quarter still differ relatively. In addition, they are relatively volatile compared to the real PD time series. With a training time series of 20 quarters or more, the predictions of *MBMA*, *BART*, *XGBoost* and *ELN* hardly differ at all. The volatility decreases and the shape of the prediction graphs resembles the shape of the graph of the real time series. Extreme PDs, i. e. local minima and maxima, are difficult to predict. This can be seen from the fact that peaks like in 2015Q3 only appear later in the predictions. The further the predicted value is from the last known PD, the less the predictions seem to describe the actual PD time series. Instead, they more closely reflect the true time series at a past point in time.
We observe similar trends in other sectors. For example, PDs for non-real-estate backed household credits show a sustained downward trend over the entire period (Figure 8). The one-step-ahead predictions of all three methods are close to the actual values and fluctuate to a similar extent. However, the times of local minima and maxima of actual and predicted values usually do not match, indicating that the exogenous variables are only of limited use for the models. As

$h$ increases, the $h$-step predictions sometimes deviate somewhat further from the true graph and lie above or below it for several consecutive quarters. There seem to be only small differences between the predictions of the different methods. These differences are more pronounced in other sectors. For the non-financial corporate sectors (Figures 16 and 18), the *MBMA* predictions deviate further from the graph of actual PDs than the other methods. Here it is particularly easy to see how an increase and subsequent decrease in true PD affects the estimates. Estimating the next twelve quarters at a point in time between the increase and decrease in PD will result in an overestimate because the decrease will not be captured by the models. The further out the forecast, the greater the overestimation.

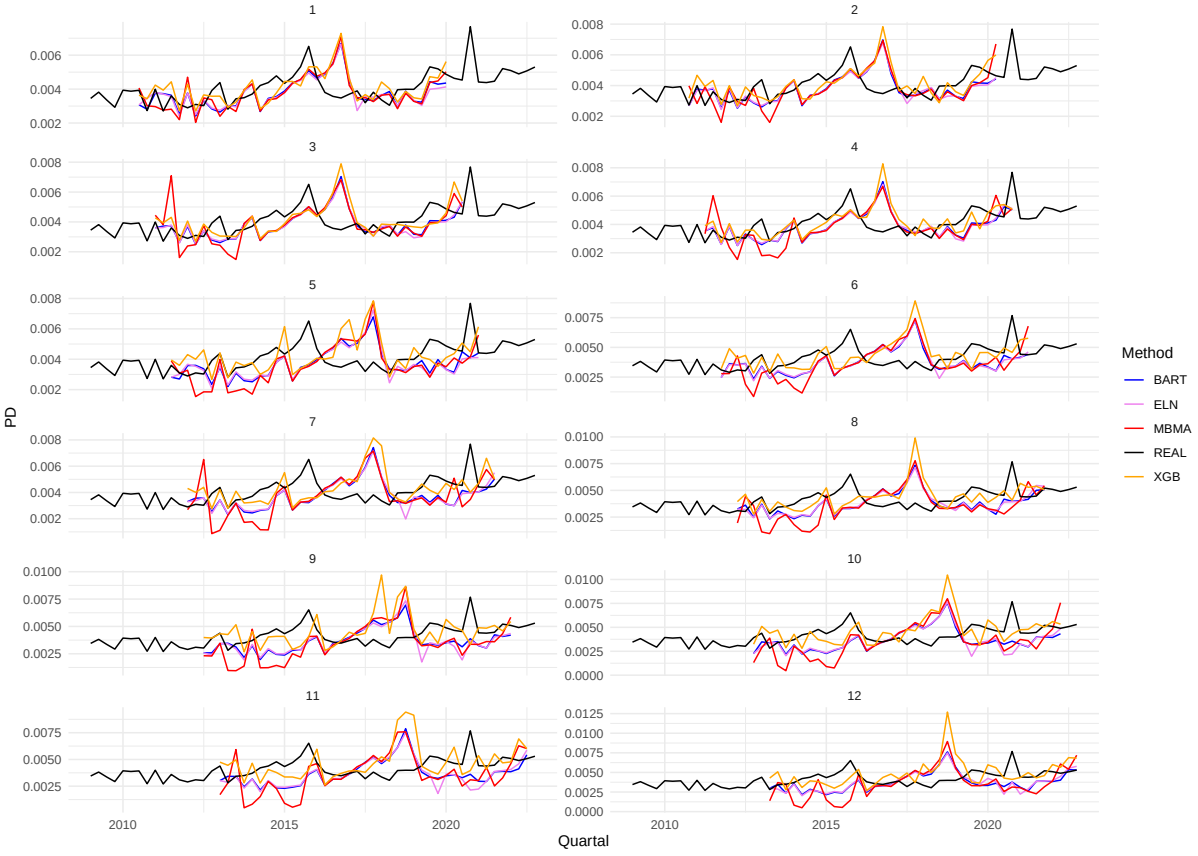


Figure 2: Forecasts of *BART, ELN, MBMA* and *XGB* plotted over the true PD time series. The number above each figure (i. e. $h$) denotes the distance (in quartals) of the forecasted value. For example, in the lower right corner we only there are only 12-step forecasts. Graph for *BMA* is left out. Sector: *Financial*. Source: Deutsche Bundesbank, Bundesbank's credit register, 2008 until 2022, Calculated by Deutsche Bundesbank (black line) and own calculation

**Forecast Horizon – Examples**

In Figure 3, we see the forecasts for the next 12 quarters from the time points 2010Q4, 2013Q4, 2016Q4 and 2019Q4. In the first period which ends in 2013Q4 the estimates of *MBMA* are higher than those of *BART*, *XGBoost* and *ELN*. Here, the underlying training time series is very short and no method can produce estimates that come close to the real time series. In the second period, it is very interesting to note that all estimated time series have peaks at the same

points in time although there are no peaks in the real time series. However, the peaks are much smaller for *MBMA*, which is why only *MBMA* can approximate the real time series. *XGBoost*, *BART* and *ELN* produce extremely high estimates here that do not match the PD data at all. In the third period, all estimated time series have a similar shape and are not far from the real time series. Especially the *MBMA* and *ELN* forecasts are very close to each other. In the fourth period, however, in which the PD suddenly rises sharply and then falls back to its previous level after a year, there are major differences. The *ELN* and *BART* predictions fluctuate around the same level. The *MBMA* and *XGBoost* models predict a significant increase in PD. With *MBMA*, this increase occurs too early and falls back to a level similar to the true time series toward the end of the period. With *XGBoost*, the increase occurs too late and therefore does not decline within the forecast period. However, this forecast is the only one that reaches the same level as the true PD. Overall the predictions in the last time span are more volatile than in the previous one. This indicates that there is at least a correlation between the explanatory variables of the models and the PDs.

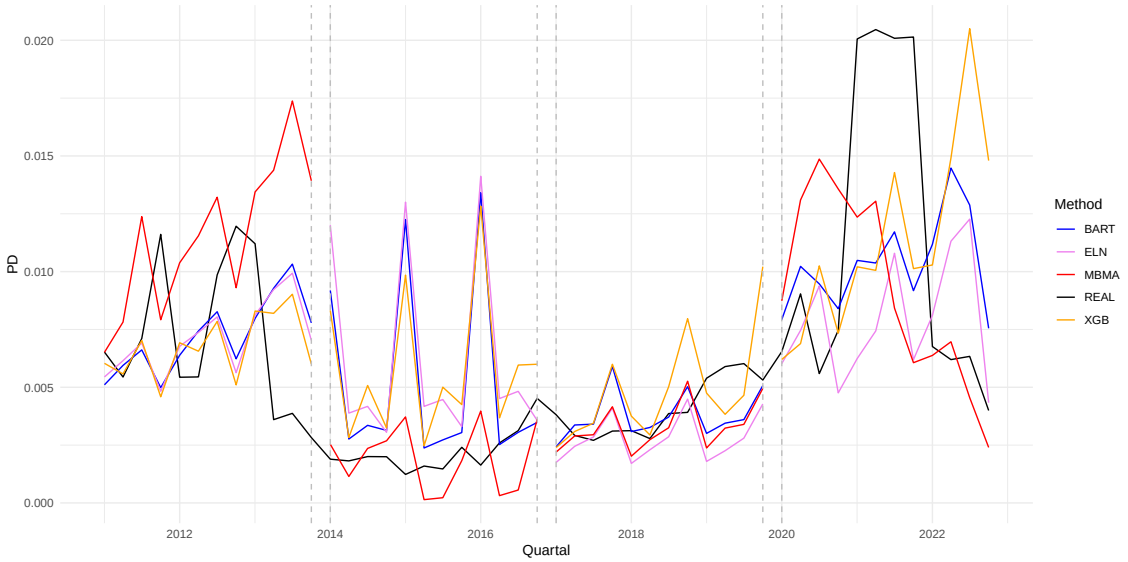


Figure 3: Forecasts of *BART, ELN, MBMA and XGB* plotted over the true PD time series. Forecasts are presented for a 12-quarter time horizon starting in 2010Q4, 2013Q4, 2016Q4, and 2019Q4 (between the vertical dotted lines). Sector: *Sov.* Source: Deutsche Bundesbank, Bundesbank's credit register, 2008 until 2022, Calculated by Deutsche Bundesbank (black line) and own calculation

The graphs for the data on the other sectors of the economy can be found in the appendix. Figure 15 on PDs for household loans secured by real estate (*HH-RE*) is particularly interesting. Here, all methods are able to predict the true time series well at certain points in time. *BART* and *ELN* perform very well, while *MBMA* and *XGBoost* overestimate the PDs.

### 4.3.2 Scores

Now, we analyze the residuals of the predictions, using the different consistent scoring functions described in Section 4.1. To maintain consistency in the scoring functions, the estimates must not be converted back into plausible default probabilities. Instead, we will consider the original

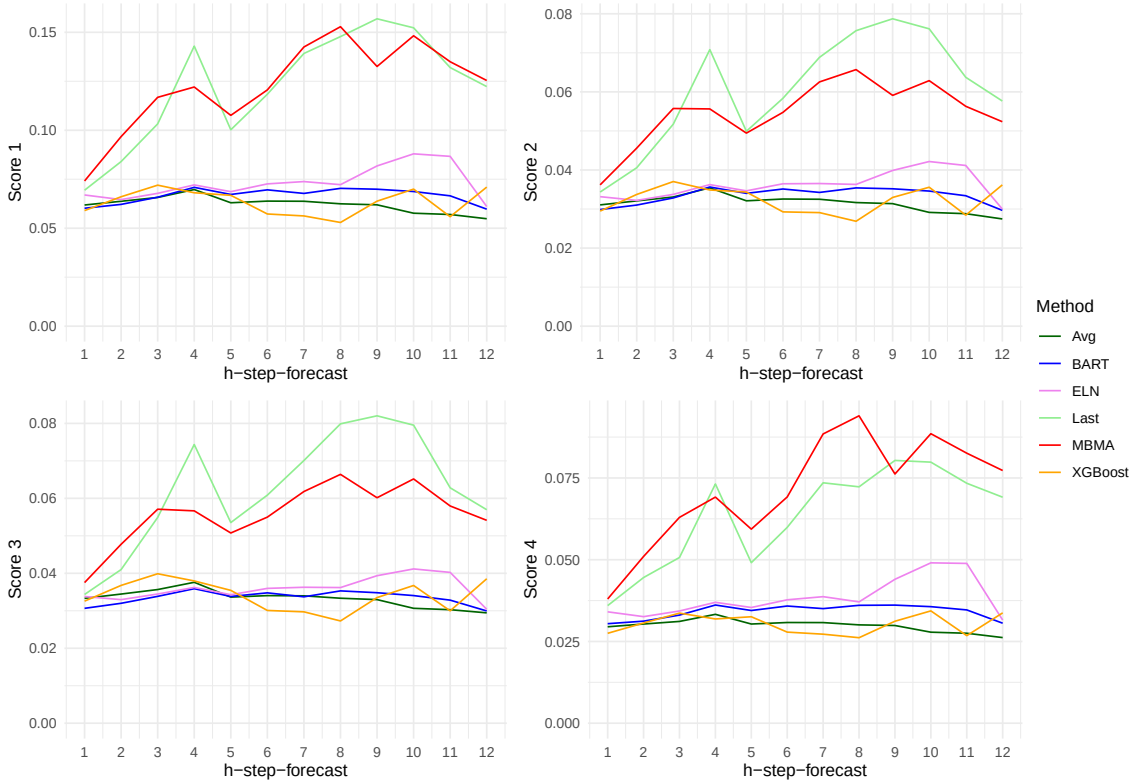


Figure 4: Average score grouped by $h$-step forecast under four different scoring function. The scoring functions are described in detail in Section 4.1. The mean squared error is shown at the top left, and the mean exponential loss is shown at the top right. The other scoring functions describe the Linear Exponential Loss for $a = 1$ and $a = -1$. Score 3 is higher for overestimated and Score 4 is higher for underestimated values. All scoring functions are consistent for the mean functional. Sector: *Financial*. Source: Deutsche Bundesbank, Bundesbank's credit register, 2008 until 2022, Own calculation

delta logit PD predictions $\hat{y}$. Again, let $y_1, \ldots, y_t$ for $t = 4, \ldots, 42$ denote the training time series of Delta-Logit-Default Probabilities and $\hat{y}_t^h$, $h = 1, \ldots, 12$ the forecast time series for the following twelve quarters $y_{t+h}$ for each $t$. Figure 4 shows graphs of four different scores. The corresponding scoring functions are:

- Score 1: $S_1(\hat{y}, y) = (\hat{y} - y)^2$ (Squared Loss)
- Score 2: $S_2(\hat{y}, y) = \exp(y) - \exp(\hat{y}) - \exp(\hat{y})(y - \hat{y})$ (Exponential Loss)
- Score 3: $S_{3+}(\hat{y}, y) = \exp(\hat{y} - y) - (\hat{y} - y) - 1$ (LinEx Loss, $a = 1$)
- Score 4: $S_{3-}(\hat{y}, y) = \exp(-(\hat{y} - y)) + (\hat{y} - y) - 1$ (LinEx Loss, $a = -1$)

We can see twelve scores per method, each for four different scoring functions. The value at position $h$ on the x-axis shows the average score of the predictions of all models for the $h$-th value after the last known PD. So here, the average is taken over the $h$-step predictions of all 39 models with varying training time series length.

$$S_s^h = \frac{1}{39} \sum_{t=4}^{42} S_s(\hat{y}_t^h, y_{t+h}), \qquad h = 1, \ldots, 12, \qquad s \in \{1, 2, 3+, 3-\} \tag{4.4}$$

We compare the scores of the methods explained above with each other and also with the scores of two simple prediction methods. For one, we use the last known value as an estimate (*Last*):

$$\hat{y}_t^h := y_t, \qquad h = 1, \ldots, 12, \qquad t = 4, \ldots, 42$$

For the second estimate, we calculate the average of all previous known values (*Avg*):

$$\hat{y}_t^h := \frac{1}{t} \sum_{i=1}^{t} y_i, \qquad h = 1, \ldots, 12, \qquad t = 4, \ldots, 42$$

Since the next 12 quarters are estimated from each point in time, these values are constant.
Let us first look at the results for the financial institutions sector as an example. The remaining charts can be found in the appendix. In principle, the *XGBoost*, *Avg*, *ELN* and *BART* methods seem to be better than the *MBMA* and *Last* method under all four scoring functions because they have much lower score values. Their scores are slightly lower at $h = 1$ and remain at a lower level as $h$ increases. The scores of *Last* and *MBMA* increase with $h$ and perform visibly worse. The choice of the scoring function has a greater effect on the result here. While the *MBMA* method under $S_{3+}$ performs better than the simple estimation method *Last* for almost all $h = 1, \ldots, 12$, its scores rise significantly under $S_{3-}$, while the *Last* scores tend to fall slightly. This suggests that *MBMA* tends to underestimate the Delta-Logit-PDs, while *Last* tends to overestimate them.
Figure 5 shows graphs similar to those in Figure 4. However, rather than calculating the normal average, the scores are weighted when determining the average. If an estimate is within or close to a risk period, the residual can be included in the average up to twelve times. Section 4.2 explains this in more detail. While the scores of *XGBoost*, *BART*, *ELN*, and *Avg* stay almost constant over time across all scoring functions, *MBMA*'s scores increase significantly. This clearly shows that *MBMA* performs the worst among the methods presented here, particularly during times of crisis. *BART*, *XGBoost*, and *Avg* perform best with data from the sector *Financial*. This result is independent of the choice of scoring function. Please note that interpreting the meaning of over- and underestimation is quite difficult since we are talking about Delta-Logit-PDs in this subsection.
Similar results were found in other economic sectors. *MBMA* and *Last* have higher scores, while the scores of the other methods are relatively close to each other. However, the scores tend to increase when the forecast is further in the future (i. e., a higher $h$ value). Figures 22 and 23 for the sovereign sector show that scores increase more with rising $h$ when predictions in crisis periods are weighted more heavily. Unlike in the financial sector, the risk-weighted scores of *MBMA* and *Last* are nearly equal to the scores of the other methods. *MBMA* does not appear to be inferior in exceptional situations.
The comparison between *BART* and *XGBoost* is interesting in the *HH-nonRE* sector (Figures 20 and 21). Here, *BART* has a lower score than *XGBoost* among all scoring functions. However, when examining the risk-weighted scores, the outcome is reversed.
In almost all sectors, the simple estimation method, *Avg*, is one of the most effective. In the *NFC-nonRE* sector (Figures 30 and 31), however, *Avg* yields poorer estimates. *Last* and *BART* are

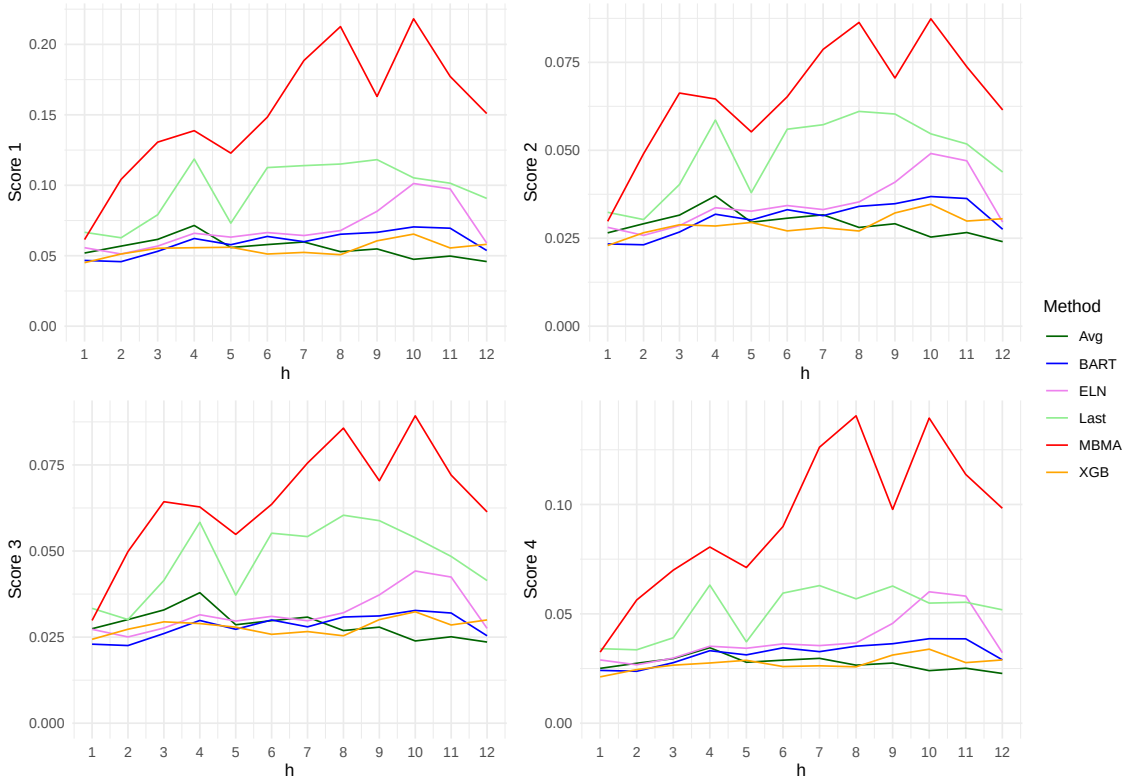


Figure 5: Medium score grouped by $h$-step forecast. This graph is similar to Graph 4. The difference is that here, the score value is calculated as the average of a weighted list of scores. The weighting is based on defined crisis periods and is described in more detail in Section 4.2. Sector: *Financial*. Source: Deutsche Bundesbank, Bundesbank's credit register, 2008 until 2022, Own calculation

the best methods here instead. Comparing scores 3 and 4 shows that *Avg* mainly overestimates the delta logit PDs. This is primarily because of the nearly monotonically falling true time series. As an average of past values, the *Avg* estimate is usually too high.

### 4.3.3 Boxplots

Now let's take a closer look at the distribution of the residuals. To do this, we will examine the Delta-Logit-PDs transformed back into plausible PDs using 4.2 and 4.3. Figure 6 shows box plots of the residuals $\varepsilon = PD - \widehat{PD}$ for each forecast method and for each $h \in \{1, \ldots, 12\}$. The box encompasses the middle 50% of the residuals. The thick black line inside the box indicates the median.

Notably, for $h = 1$, the boxes of all methods are approximately the same size. However, for larger $h$, significant differences emerge. The *BMA* box is notably larger than those of the other methods, and the *MBMA* box is slightly larger as well. This indicates that the residuals from *MBMA* and, more notably, *BMA* are more widely scattered, meaning the predictions deviate more frequently from the true value. Additionally, the medians of *BART*, *BMA*, *ELN*, and *MBMA* are above zero for all $h$, except for *MBMA*, where some PD estimates are much too high. This also applies to most of the middle 50% of the residuals. These methods therefore underestimate the PDs much more often than they overestimate them. The situation is different

for the residuals of *XGBoost*. Here, the median is close to zero for most $h$, showing that *XGBoost* underestimates and overestimates the PDs with approximately equal frequency.
However, these results vary by economic sector. For countries with high risk ratings (Figure 34), both *BMA* and *MBMA* often produce very high estimates, as reflected by many high negative residuals. In contrast, for countries with low risk ratings (Figure 33), the box plots for all methods are similar across the entire forecast horizon. In the real estate-backed loan sector for households (Figure 35), the median residuals from *XGBoost* are lower than those from the other methods. In this case, however, it is in negative territory, while the others are closer to zero.

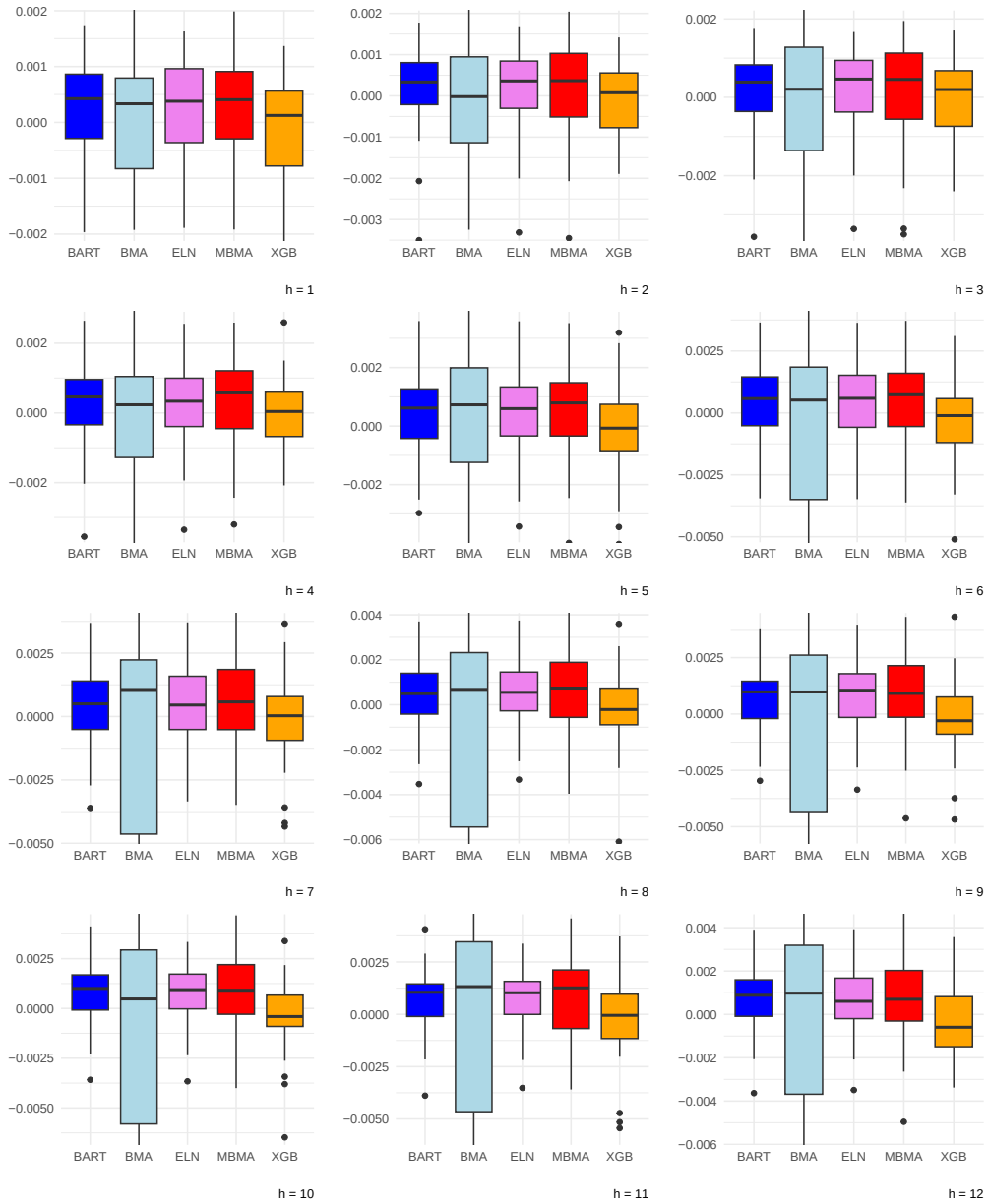


Figure 6: Boxplots of the errors $PD - \widehat{PD}$ for the $h$-step forecasts. The $h$ value is displayed to the right of each graph. For each $h = 1, \ldots, 12$, all $h$-step predictions are summarized in a graph. To keep the graphs readable, the y-axis is truncated at the $5th$ and $97.5th$ percentiles. These quantiles are calculated by summarizing the residuals of all methods per $h$-step in a vector. Sector: *Financial*. Source: Deutsche Bundesbank, Bundesbank's credit register, 2008 until 2022, Own calculation

### 4.3.4 Diebold-Mariano-Test

We also want to test whether the residuals that arise from the different methods differ significantly in statistical terms. To do this, we use a modified version of the *Diebold-Mariano test* (DM test), which was presented by Harvey et al. (1997). For implementation, we use the function **dm.test** of the R package **forecast**.
Since the Diebold-Mariano test accounts for the autocovariances of the error differences, we cannot use the $h$-step forecasts for different $h$ in a single test. Instead, we have to analyze the errors of the forecasts separately for each $h = 1, \ldots, 12$. That is, for a test we only use the errors $\varepsilon_4^h, \ldots, \varepsilon_{42}^h$ for one $h \in \{1, \ldots, 12\}$. The errors of the forecasted Delta-Logit-PDs are used in the R-method **dm.test**. We set the power parameter to two, which means we use a quadratic loss function. Additionally, we consider a test in which we evaluate all $h$-step predictions simultaneously. While this test is no longer statistically valid, the results can provide a useful overview of the methods. To accomplish this, we pass $h = 6$ to the test method as the mean distance to the predicted value. In this test, we also compare the errors of the *Last* and *Avg* simple models.
The methods are compared in pairs. We conduct a one-sided test in each case, with the null hypothesis being that the expected squared errors of both methods are equal. The alternative hypothesis states that method in the header row of the table is less accurate than the method in the column on the left side.

$$H_0 : \mathbb{E}[\varepsilon_1^2] = \mathbb{E}[\varepsilon_2^2], \qquad H_1 : \mathbb{E}[\varepsilon_1^2] < \mathbb{E}[\varepsilon_2^2]$$

$\varepsilon_1$ and $\varepsilon_2$ denote the residuals of the forecasts produced by the first and second method that are compared in the test. First, we take a look at the results of the Diebold-Mariano-Test for the sector for financial institutes. The $p$-values can be read off in Tables 2 and 15. The method on the left side of the table denotes the first method and the one in the top row denotes the second one. For example, the $p$-value of 0.01 in the top left table would indicate that we should reject the null hypothesis and assume that *BMA* is less accurate than *MBMA* instead.
Considering the 1-step and 2-step forecasts, we reject the null hypothesis of equal squared errors at a significance level of $\alpha = 1\%$ in the pairwise comparisons of *BMA* with all other methods. The alternative hypothesis, which we assume to be true, states that *BMA* produces less accurate predictions than *MBMA*, *BART*, *XGBoost* and *ELN*. For predictions further in the

| | MBMA | BMA | BART | XGB | ELN | Last | Avg |
|---|---|---|---|---|---|---|---|
| MBMA | | 0.01 | 1.00 | 1.00 | 1.00 | 0.51 | 1.00 |
| BMA | 0.99 | | 1.00 | 1.00 | 1.00 | 0.99 | 1.00 |
| BART | 0.00 | 0.00 | | 0.92 | 0.11 | 0.00 | 0.96 |
| XGB | 0.00 | 0.00 | 0.08 | | 0.03 | 0.00 | 0.45 |
| ELN | 0.00 | 0.00 | 0.89 | 0.97 | | 0.00 | 0.99 |
| Last | 0.49 | 0.01 | 1.00 | 1.00 | 1.00 | | 1.00 |
| Avg | 0.00 | 0.00 | 0.04 | 0.55 | 0.01 | 0.00 | |

Table 2: $p$-values of the pairwise Diebold-Mariano test for $h = 1, \ldots, 12$. Sector: *Financial*. Source: Deutsche Bundesbank, Bundesbank's credit register, 2008 until 2022, Own calculation

future, the $p$-values are between 0.02 and 0.13. Therefore, we cannot reject the null hypothesis at a conventional significance level $\alpha$. However, it should be kept in mind that the tests are carried out with 39 data points per method on a relatively small data set, so $\alpha$ should not be set too small. The $p$-values of just over 0.1 at least suggest that there is evidence for the superiority of *MBMA*, *BART* and *XGBoost* over *BMA*.

All $p$-values for the *BART* and *MBMA* tests are below 0.16, and six out of twelve are even below 0.1. Therefore, we can assume that *BART* performs significantly better than *MBMA* on at least the first half of the prediction horizon. The *XGBoost* and *ELN* methods seem to perform similarly and better than the *MBMA* method. The p-values for $h = 1, \dots, 6$ are usually below 0.1, while those for $h = 7, \dots, 12$ range from 0.1 to 0.2. Therefore, we can conclude that *XGBoost* and *ELN* produce significantly better estimates for the first six quarters of the prediction horizon. Interestingly, though, the one-step forecasts of *ELN* are not significantly better, with a relatively high $p$-value of 0.15. No method stands out when comparing *BART*, *XGBoost*, and *ELN* with each other.

The picture changes when we consider the residuals of all $h$-step predictions in a single test. The expected quadratic residuals of *BART*, *XGBoost* and *ELN* are significantly lower than those of *MBMA*; even on a low significance level of $\alpha = 0.1\%$. Please note that we are now considering $12 \cdot 39$ data points, so we should apply a lower significance level than before. The two simple methods, *Last* and *Avg*, are clearly superior to the *BMA*. *Avg* is also significantly better than *MBMA*, *BART* and *ELN* while *Last* is inferior to *BART*, *XGBoost* and *ELN*.

The classic *BMA* method performs the worst in all sectors. All other methods deliver significantly better results. In every other economic sector, there is at least one $h \in \{1, \dots, 12\}$ for which the predictions of *BART*, *XGBoost*, and *ELN* are better than the *MBMA* predictions. However, this is not true for the entire forecast horizon. Conversely, *MBMA* is never superior to any of the three methods. Looking at tables 16 to 21, it's clear that *MBMA* is inferior. In most cases, the simple *Avg* method produces better results than *MBMA*, and sometimes even *BART*, *XGBoost*, or *ELN*. However, as we see in the *NFC-nonRE* sector, we cannot fully rely on *Avg*. Here, the complex methods are significantly better. The simple *Last* method is usually inferior. The only conclusion that can be drawn with a high degree of certainty is that the classic *BMA* method is inferior to the others. There are also indications that *BART*, *XGBoost*, and *ELN* are better suited than *MBMA* for predicting PDs across sectors. The simple method *Avg* should also be considered.

The difference between the DM test results in the sectors *Financial* and *NFC-nonRE* two sectors is likely due to the shape of the PD time series. For the sector *NFC-nonRE*, it declines almost monotonically over the entire period which is why the average over the past values is not a good predictor. In the *Financial* sector, however, it fluctuates more. If exogenous variables are not very helpful taking the average can be a good approach.

### 4.3.5 Feature Importance

*Feature importance* indicates the significance of an explanatory variable for the model. In this section, we examine which variables these are, whether they differ depending on the model, and if there are differences between economic sectors.

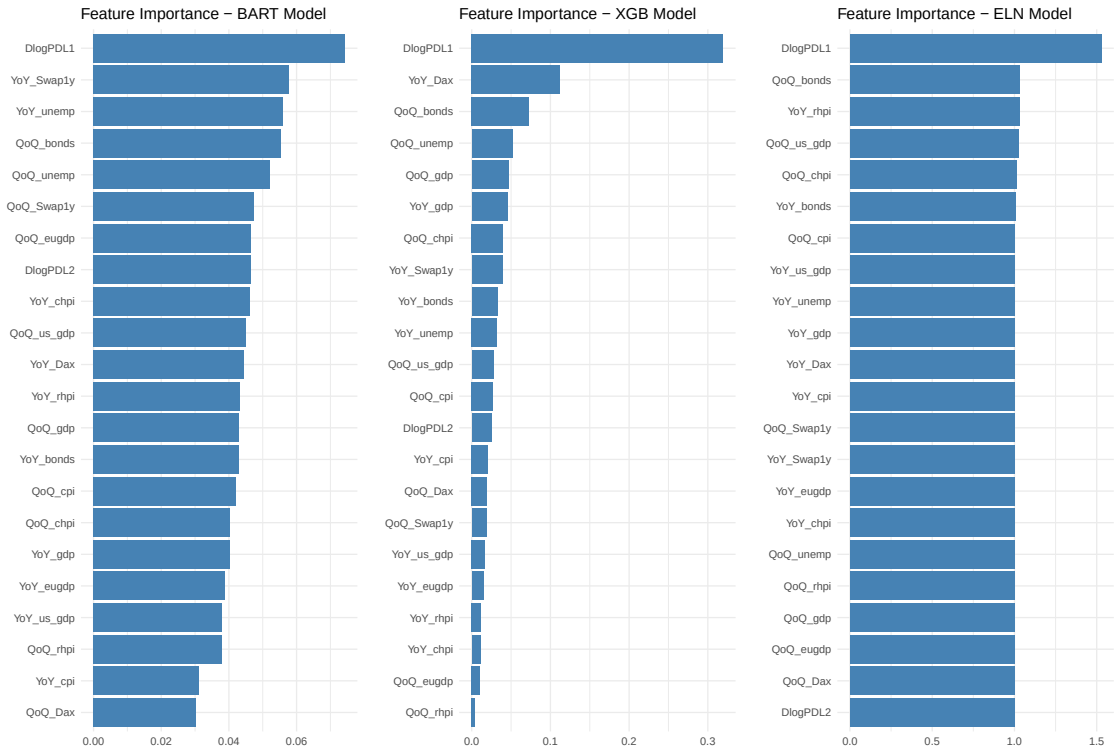


Figure 7: Feature importance of *BART*, *XGBoost* and *ELN* model. Sector: *HH-nonRE*. Source: Deutsche Bundesbank, Bundesbank's credit register, 2008 until 2022, Own calculation

Figure 7 shows the results for the sector *HH-nonRE* as an example. Values on the x-axis cannot easily be compared across different models. It is only used to compare variables within a single model. Using the *BART*, *XGBoost*, and *ELN* methods, we build a model based on all available data points; that is, the training time series is 54 long. Then, we use the method **xgb.importance** included in the R package to calculate the feature importance for each model. For the *BART* model, we calculate this manually. First, we calculate the average frequency with which the variables appear in all models. We then use the relative average frequency as feature importance. For the *ELN* model, we use the R package **IML**. It permutes the values of all variables individually to remove their correlation with the endogenous variable. The feature importance is derived from the increase in prediction error. For more on that, see Fisher et al. (2019).

Clearly, the lagged endogenous variable *DlogPDL1* is the most important for all models. Otherwise, the models are not similar. Generally, the importance of the exogenous variables in the *BART* and *ELN* models is similar. Thus, no macroeconomic variable is particularly well-suited for predicting PDs. This seems to be somewhat different for the *XGBoost* model, however. However, since it is probabilistic, the values may change if a new model is trained on the same data with the same hyperparameters. After repeated testing, only the statement in the graph regarding *DlogPDL1* appears reliable.

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.01 | 0.95 | 0.96 | 0.85 |
| BMA | 0.99 | | 0.99 | 1.00 | 0.99 |
| BART | 0.05 | 0.01 | | 0.82 | 0.08 |
| XGB | 0.04 | 0.00 | 0.18 | | 0.06 |
| ELN | 0.15 | 0.01 | 0.92 | 0.94 | |

Table 3: 1-step forecast

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.01 | 0.97 | 0.97 | 0.97 |
| BMA | 0.99 | | 0.99 | 0.99 | 0.99 |
| BART | 0.03 | 0.01 | | 0.72 | 0.45 |
| XGB | 0.03 | 0.01 | 0.28 | | 0.25 |
| ELN | 0.03 | 0.01 | 0.55 | 0.75 | |

Table 4: 2-step forecast

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.04 | 0.96 | 0.99 | 0.95 |
| BMA | 0.96 | | 0.98 | 0.98 | 0.98 |
| BART | 0.04 | 0.02 | | 0.92 | 0.50 |
| XGB | 0.01 | 0.02 | 0.08 | | 0.06 |
| ELN | 0.05 | 0.02 | 0.50 | 0.94 | |

Table 5: 3-step forecast

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.07 | 0.90 | 0.86 | 0.90 |
| BMA | 0.93 | | 0.95 | 0.95 | 0.95 |
| BART | 0.10 | 0.05 | | 0.42 | 0.75 |
| XGB | 0.14 | 0.05 | 0.58 | | 0.68 |
| ELN | 0.10 | 0.05 | 0.25 | 0.32 | |

Table 6: 4-step forecast

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.07 | 0.94 | 0.86 | 0.94 |
| BMA | 0.93 | | 0.95 | 0.95 | 0.95 |
| BART | 0.06 | 0.05 | | 0.30 | 0.33 |
| XGB | 0.14 | 0.05 | 0.70 | | 0.64 |
| ELN | 0.06 | 0.05 | 0.67 | 0.36 | |

Table 7: 5-step forecast

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.11 | 0.90 | 0.91 | 0.89 |
| BMA | 0.89 | | 0.92 | 0.92 | 0.92 |
| BART | 0.10 | 0.08 | | 0.88 | 0.11 |
| XGB | 0.09 | 0.08 | 0.12 | | 0.09 |
| ELN | 0.11 | 0.08 | 0.89 | 0.91 | |

Table 8: 6-step forecast

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.13 | 0.88 | 0.89 | 0.89 |
| BMA | 0.87 | | 0.90 | 0.90 | 0.90 |
| BART | 0.12 | 0.10 | | 0.84 | 0.42 |
| XGB | 0.11 | 0.10 | 0.16 | | 0.13 |
| ELN | 0.11 | 0.10 | 0.58 | 0.87 | |

Table 9: 7-step forecast

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.12 | 0.87 | 0.86 | 0.87 |
| BMA | 0.88 | | 0.90 | 0.90 | 0.90 |
| BART | 0.13 | 0.10 | | 0.66 | 0.51 |
| XGB | 0.14 | 0.10 | 0.34 | | 0.35 |
| ELN | 0.13 | 0.10 | 0.49 | 0.65 | |

Table 10: 8-step forecast

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.11 | 0.86 | 0.86 | 0.86 |
| BMA | 0.89 | | 0.91 | 0.92 | 0.91 |
| BART | 0.14 | 0.09 | | 0.86 | 0.22 |
| XGB | 0.14 | 0.08 | 0.14 | | 0.13 |
| ELN | 0.14 | 0.09 | 0.78 | 0.87 | |

Table 11: 9-step forecast

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.11 | 0.84 | 0.82 | 0.85 |
| BMA | 0.89 | | 0.92 | 0.92 | 0.92 |
| BART | 0.16 | 0.08 | | 0.67 | 0.35 |
| XGB | 0.18 | 0.08 | 0.33 | | 0.33 |
| ELN | 0.15 | 0.08 | 0.65 | 0.67 | |

Table 12: 10-step forecast

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.10 | 0.85 | 0.80 | 0.84 |
| BMA | 0.90 | | 0.91 | 0.91 | 0.91 |
| BART | 0.15 | 0.09 | | 0.44 | 0.01 |
| XGB | 0.20 | 0.09 | 0.56 | | 0.43 |
| ELN | 0.16 | 0.09 | 0.99 | 0.57 | |

Table 13: 11-step forecast

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.10 | 0.85 | 0.81 | 0.86 |
| BMA | 0.90 | | 0.91 | 0.91 | 0.91 |
| BART | 0.15 | 0.09 | | 0.43 | 0.38 |
| XGB | 0.19 | 0.09 | 0.57 | | 0.53 |
| ELN | 0.14 | 0.09 | 0.62 | 0.47 | |

Table 14: 12-step forecast

Table 15: $p$-values of the pairwise Diebold-Mariano test - individual tests for $h$-step forecasts for each $h \in \{1, \ldots, 12\}$. Sector: *Financial*. Source: Deutsche Bundesbank, Bundesbank's credit register, 2008 until 2022, Own calculation

# 5 Conclusion

In this study, we fitted various regression-based, tree-based, and simple models to a dataset of default probabilities and macroeconomic variables to predict the future values of PD time series. We then compared the results using visual comparisons of forecasts and error distributions, scores of multiple scoring functions, and the Diebold-Mariano test. All of these comparisons revealed that the classical Bayesian Model Averaging approach is inferior to the other methods. The squared forecast errors are much higher here, as can clearly be seen in the DM test result tables. However, the results are not that clear otherwise. When we compared the scores of the $h$-step forecasts, the simple forecasting method *Avg* was one of the best methods for most sectors. Since *Avg* is not based on macroeconomic variables, they don't seem to explain much of the dependent variable. The analysis of feature importance supports this conclusion, as we could not identify a single variable that consistently remained relevant across different models. Regarding the scores, however, the *BART*, *XGBoost*, and *ELN* models perform as well as the *Avg* models. The analysis of the NFC-nonRE sector showed that the simple model is not suitable for all datasets; therefore, a more complex model is preferable. In most cases, there is not much difference, but in some cases, it can be useful not to rely solely on past values of the dependent variable.
The scores, boxplots of forecast errors, and Diebold-Mariano test indicate that Modified Bayesian Model Averaging, the method that most closely resembles the current benchmark, outperformes classical Bayesian Model Averaging but underperforms compared to other regression- and tree-based methods. Weighted scores show that *MBMA* forecasts can be even more unreliable during periods of economic crisis.
The results of the DM tests typically align with the scores obtained. For instance, Figures 22 and 23 show that the *XGBoost* models produce the lowest scores most of the time. However, the difference seems to be not that significant. It is therefore interesting that, according to the Diebold-Mariano test, XGB errors are lower than all other errors at a significance level of $\alpha = 1\%$. The results of DM test and Scores also coincide with regard to the inferiority of *MBMA* compared to other processes. However, it appears that the DM test detects differences that are barely visible in the plots.

The results differ across different economic sectors, showing that there is no single forecasting method that is preferable in the context of general credit risk stress testing. The most important limitation of our analysis is the shortness of the PD time series. Only 60 data points are available because PD measurement began in 2008. The training time series is even shorter due to the 12-quarter forecast horizon and the deltas we apply. The problem of insufficient training and validation data becomes even more significant when we are specifically interested in forecast quality during adverse scenarios. Although the ESRB identifies some crisis quarters between 2008 and 2022, we do not observe extreme PD increases in every sector. This means there is almost no crisis data on which we can train or test our models.
Our Modified Bayesian Model Averaging model does not contain a benchmark constraint and is therefore not completely identical to the *BCBMA* model, which is currently the benchmark

model. But the *MBMA* model is inferior to the *BART*, *XGBoost*, and *ELN* models, what can at leat give some hints on the properties and the behaviour of *BCBMA*. At the edges of distributions, where a predefined adverse macroeconomic scenario would be located, parametric models, such as linear models, typically provide more reliable estimations than non-parametric models. In contrast, tree-based models are more effective in regions with many data points. Unfortunately, we cannot empirically validate this generalization because we lack testing data on the edges. However, the weighted scores do not indicate such behavior. Even if one prefers a regression-based model to nonparametric models, this research suggests choosing the Elastic Net.

In a much larger dataset, it could be interesting to train and compare a neural network. Unfortunately, this proved to be ineffective with fewer than 60 data points. Another solution could be to search for additional regressors that help explain the default probabilities. If more adverse scenario PD data becomes available in the future, one should pay more attention to those forecasts.

# A Forecast Graphics

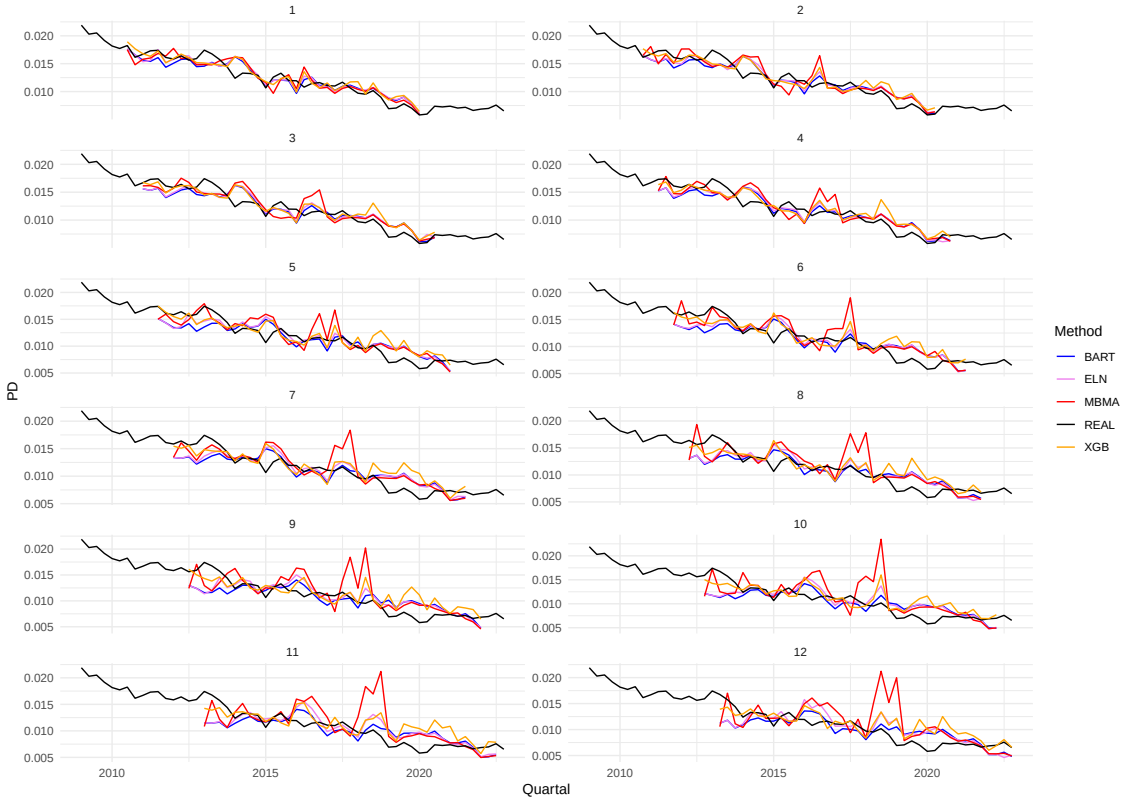


Figure 8: Compare for description of Figure 2. Sector: *HH-nonRE*. Source: Deutsche Bundesbank, Bundesbank's credit register, 2008 until 2022, Calculated by Deutsche Bundesbank (black line) and own calculation

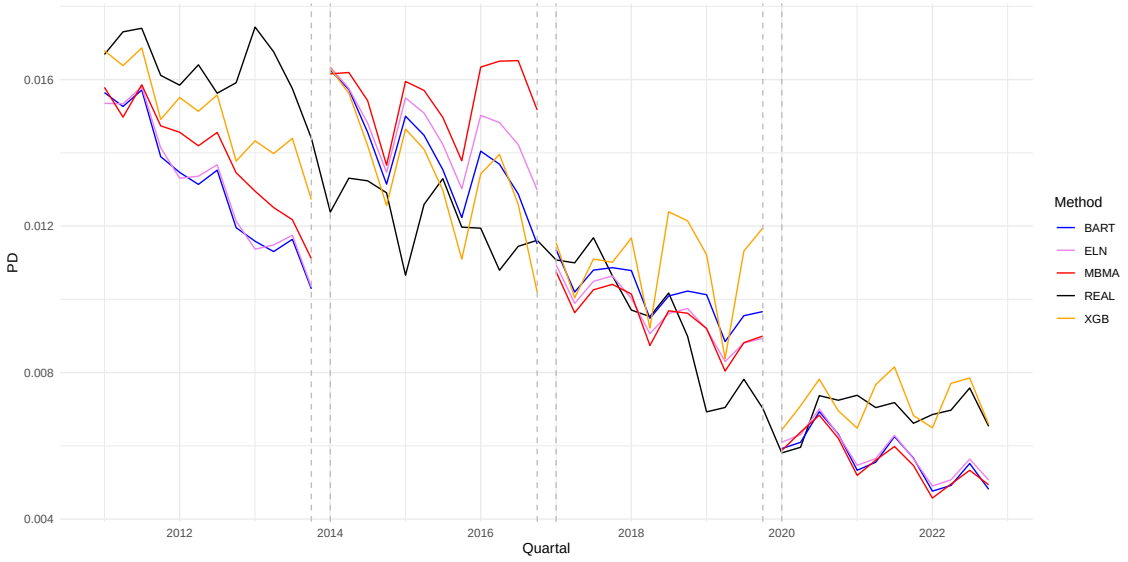


Figure 9: Compare for description of Figure 3. Sector: *HH-nonRE*. Source: Deutsche Bundesbank, Bundesbank's credit register, 2008 until 2022, Calculated by Deutsche Bundesbank (black line) and own calculation

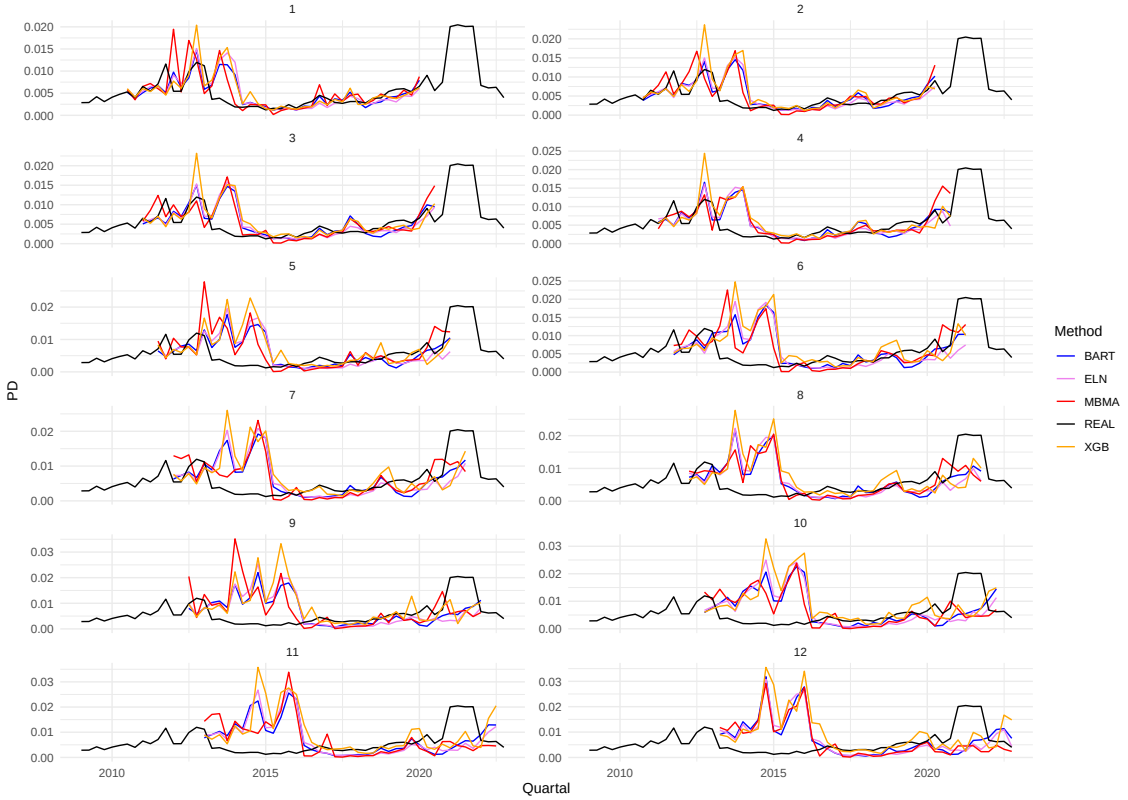


Figure 10: Compare for description of Figure 2. Sector: *Sov.* Source: Deutsche Bundesbank, Bundesbank's credit register, 2008 until 2022, Calculated by Deutsche Bundesbank (black line) and own calculation

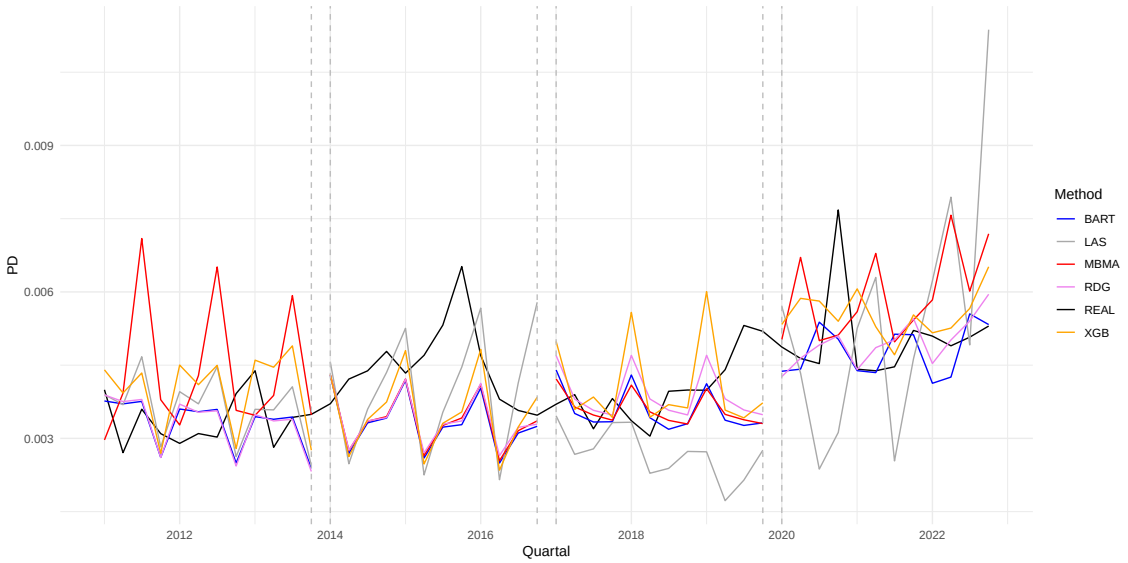


Figure 11: Compare for description of Figure 3. Sector: *Financial.* Source: Deutsche Bundesbank, Bundesbank's credit register, 2008 until 2022, Calculated by Deutsche Bundesbank (black line) and own calculation

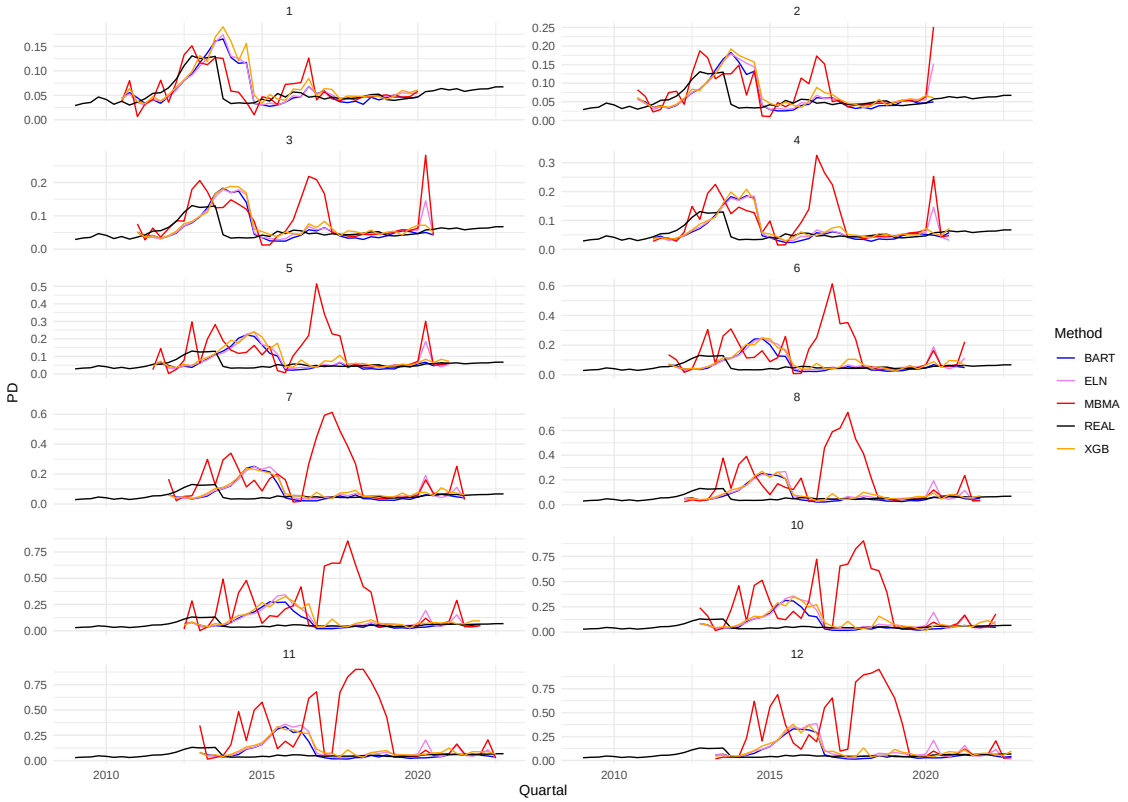


Figure 12: Compare for description of Figure 2. Sector: *Sov-HR*. Source: Deutsche Bundesbank, Bundesbank's credit register, 2008 until 2022, Calculated by Deutsche Bundesbank (black line) and own calculation

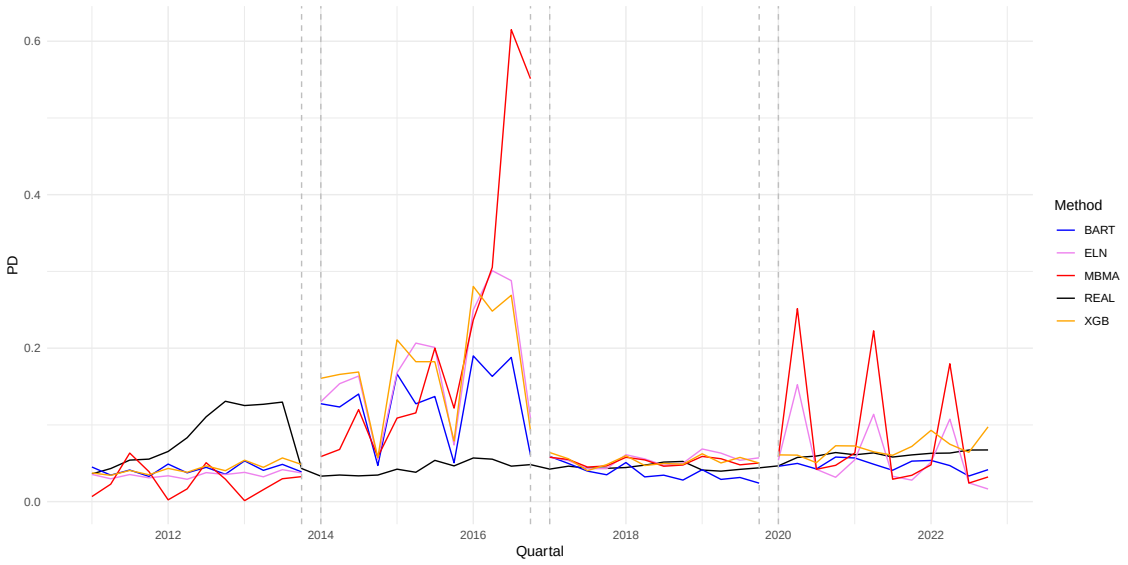


Figure 13: Compare for description of Figure 3. Sector: *Sov-HR*. Source: Deutsche Bundesbank, Bundesbank's credit register, 2008 until 2022, Calculated by Deutsche Bundesbank (black line) and own calculation

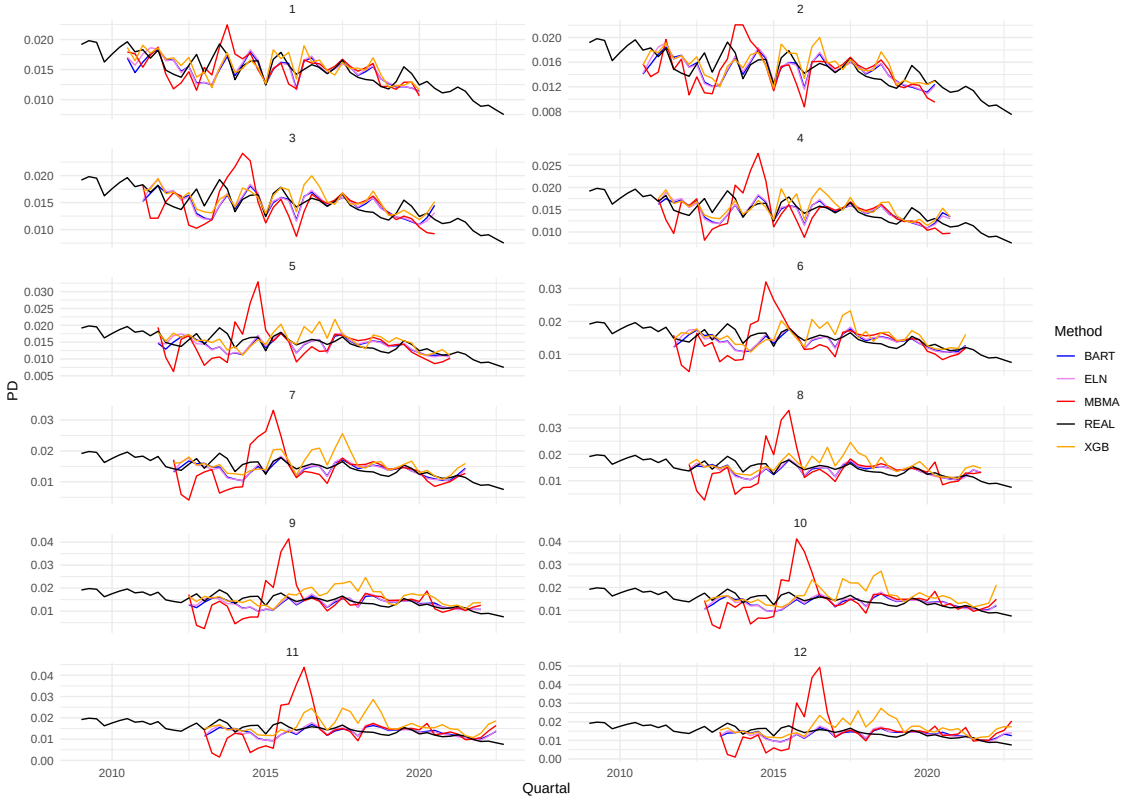


Figure 14: Compare for description of Figure 2. Sector: *HH-RE*. Source: Deutsche Bundesbank, Bundesbank's credit register, 2008 until 2022, Calculated by Deutsche Bundesbank (black line) and own calculation

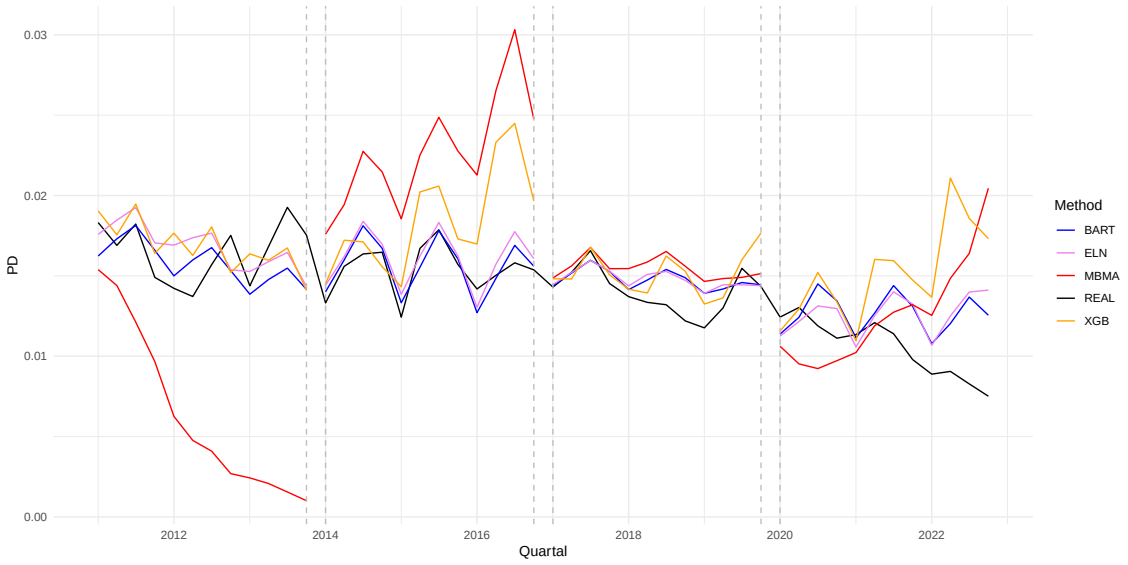


Figure 15: Compare for description of Figure 3. Sector: *HH-RE*. Source: Deutsche Bundesbank, Bundesbank's credit register, 2008 until 2022, Calculated by Deutsche Bundesbank (black line) and own calculation

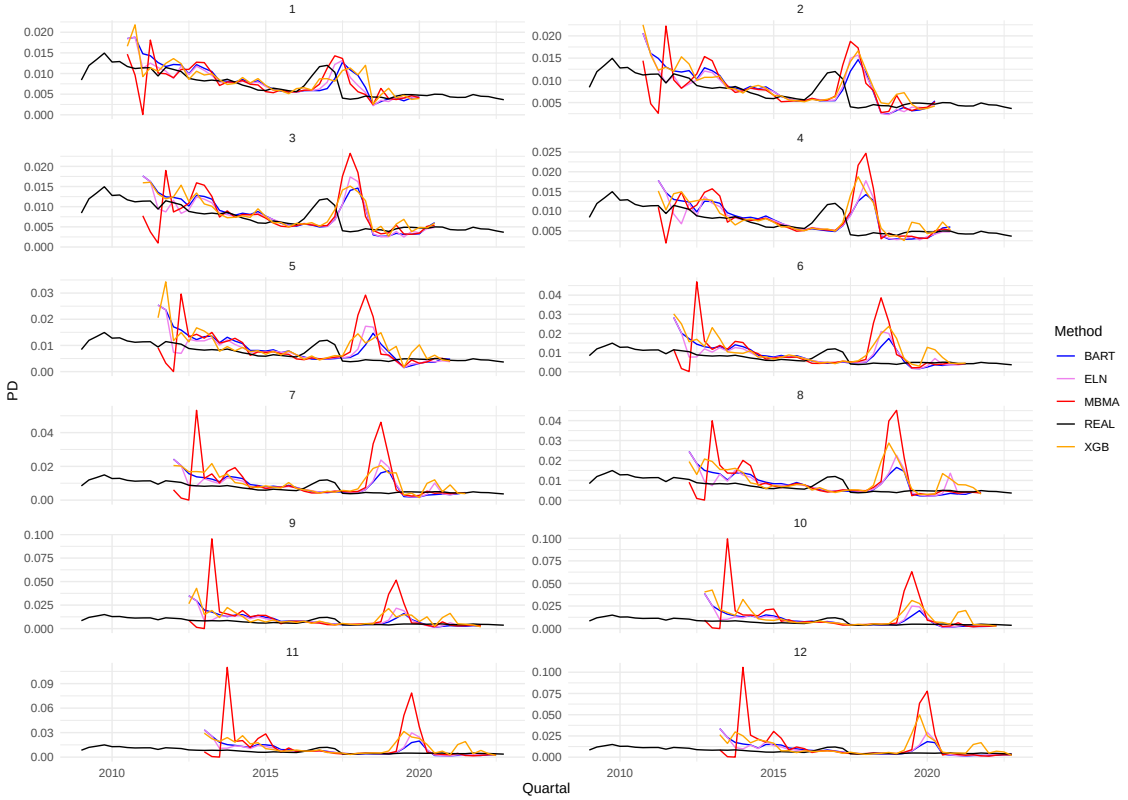


Figure 16: Compare for description of Figure 2. Sector: *NFC-RE*. Source: Deutsche Bundesbank, Bundesbank's credit register, 2008 until 2022, Calculated by Deutsche Bundesbank (black line) and own calculation

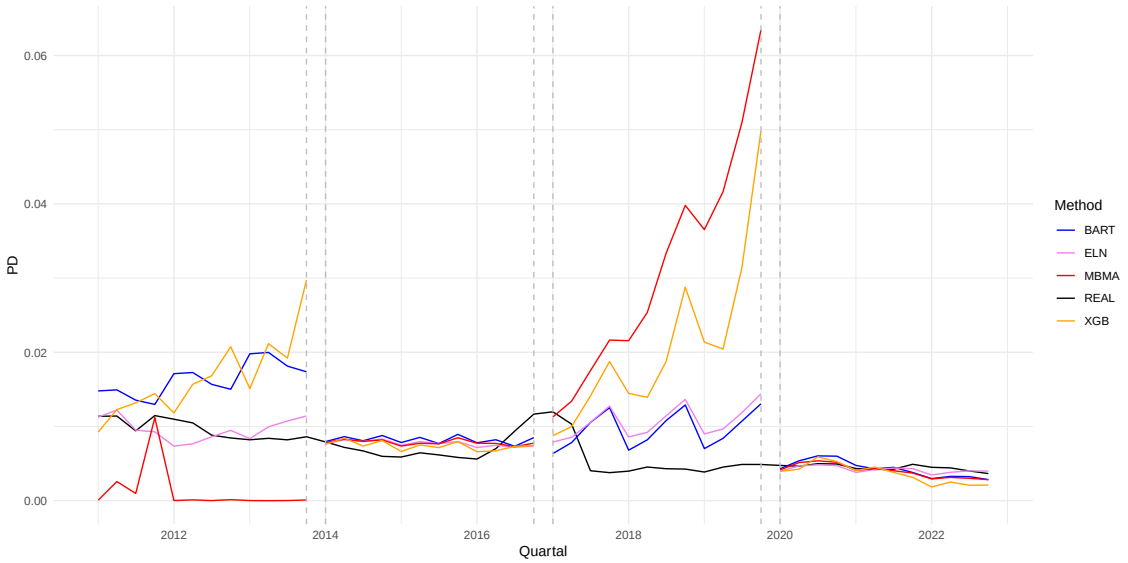


Figure 17: Compare for description of Figure 3. Sector: *NFC-RE*. Source: Deutsche Bundesbank, Bundesbank's credit register, 2008 until 2022, Calculated by Deutsche Bundesbank (black line) and own calculation

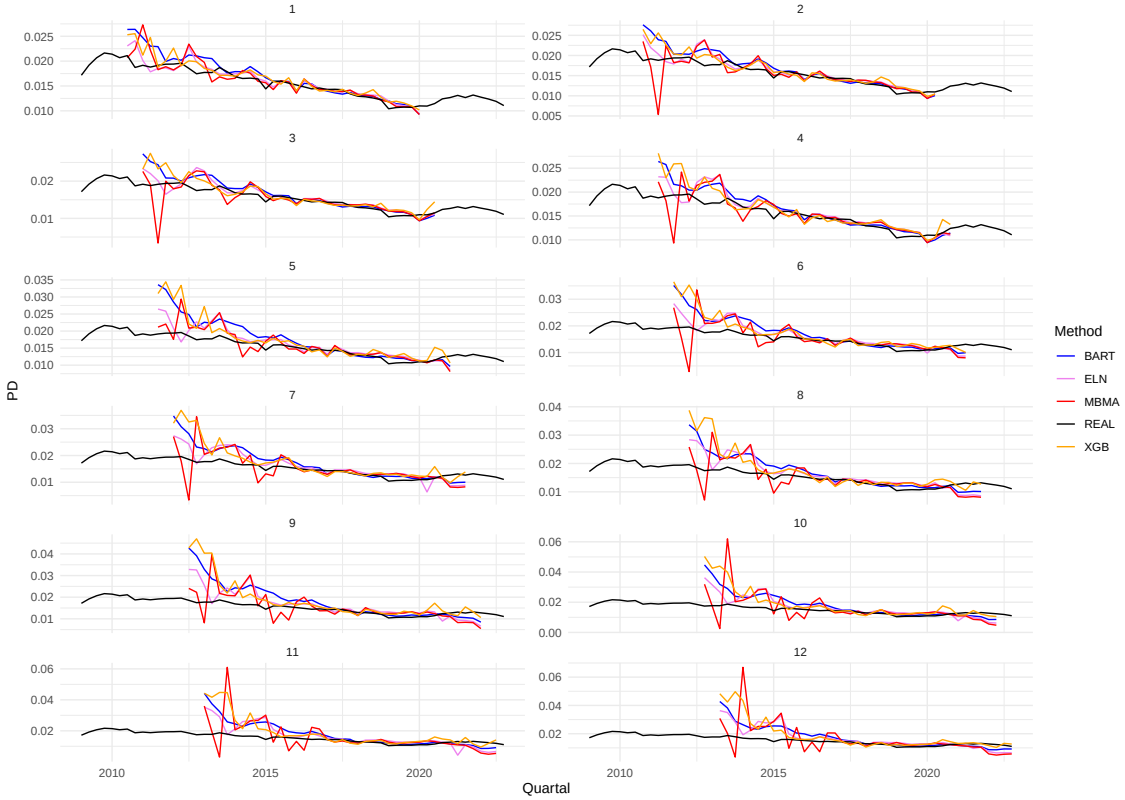


Figure 18: Compare for description of Figure 2. Sector: *NFC-nonRE*. Source: Deutsche Bundesbank, Bundesbank's credit register, 2008 until 2022, Calculated by Deutsche Bundesbank (black line) and own calculation

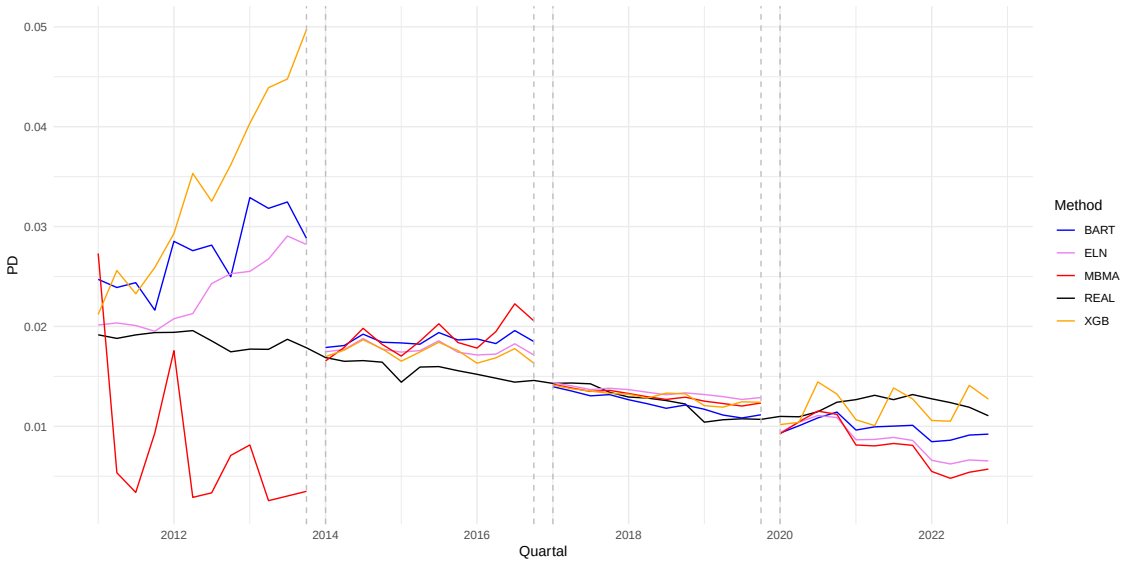


Figure 19: Compare for description of Figure 3. Sector: *NFC-nonRE*. Source: Deutsche Bundesbank, Bundesbank's credit register, 2008 until 2022, Calculated by Deutsche Bundesbank (black line) and own calculation

# B Scores

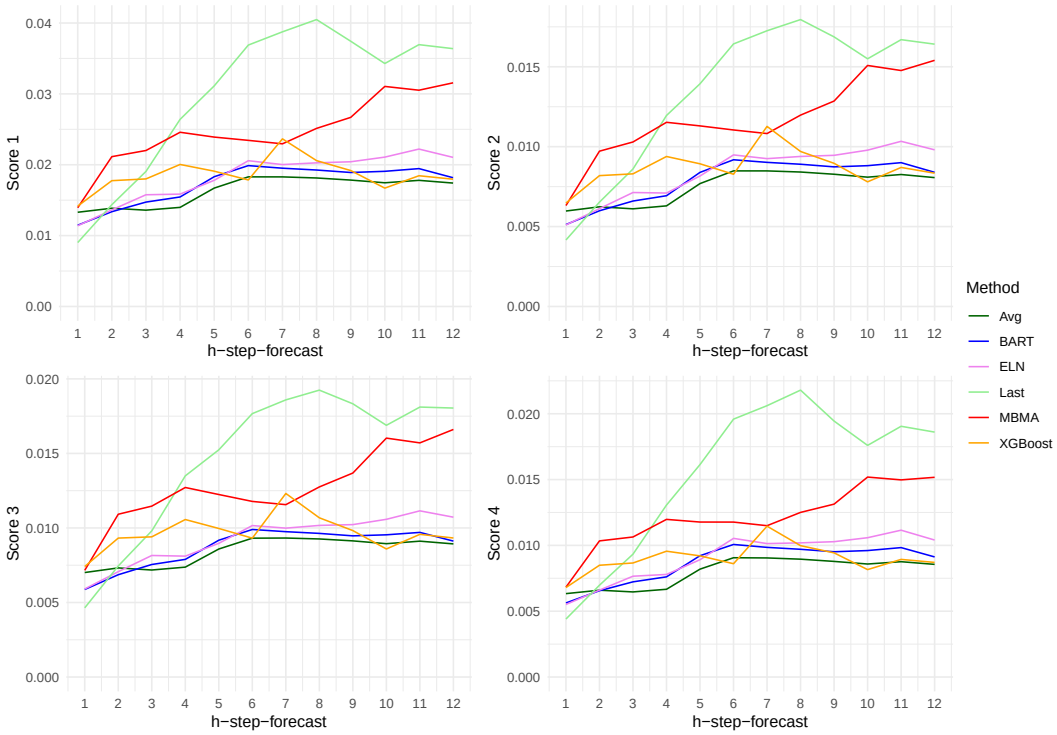


Figure 20: Compare for description of Figure 4. Sector: *HH-nonRE*. Source: Deutsche Bundesbank, Bundesbank's credit register, 2008 until 2022, Own calculation


Figure 21: Compare for description of Figure 5. Sector: *HH-nonRE*. Source: Deutsche Bundesbank, Bundesbank's credit register, 2008 until 2022, Own calculation

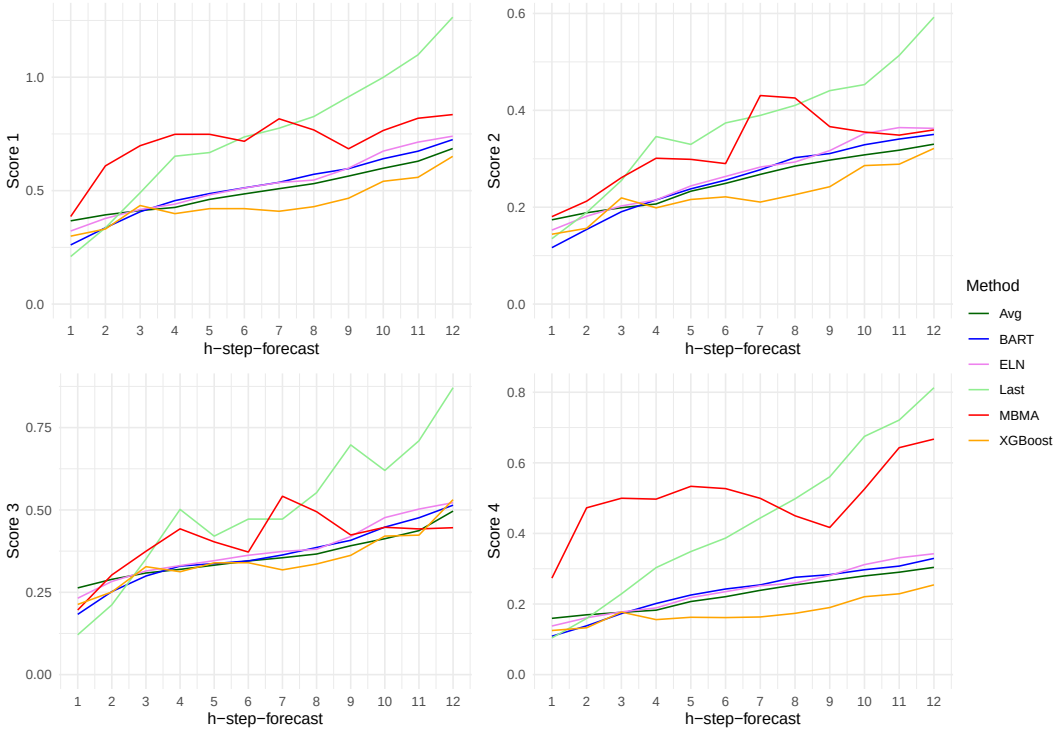


Figure 22: Compare for description of Figure 4. Sector: *Sov.* Source: Deutsche Bundesbank, Bundesbank's credit register, 2008 until 2022, Own calculation

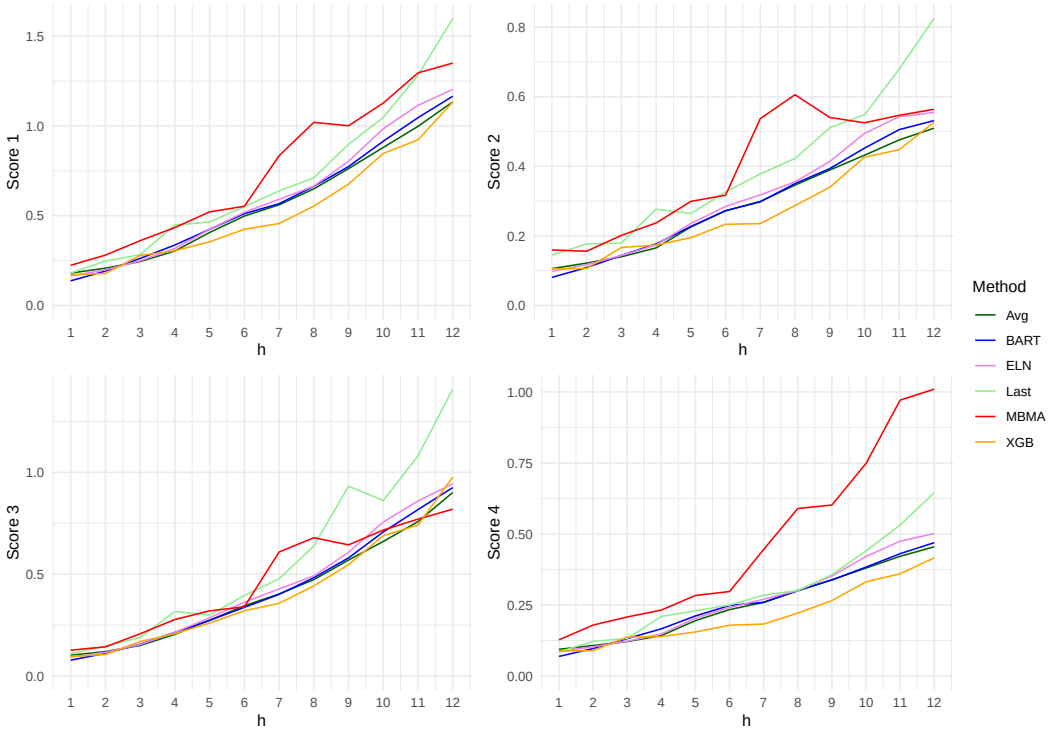


Figure 23: Compare for description of Figure 5. Sector: *Sov.* Source: Deutsche Bundesbank, Bundesbank's credit register, 2008 until 2022, Own calculation

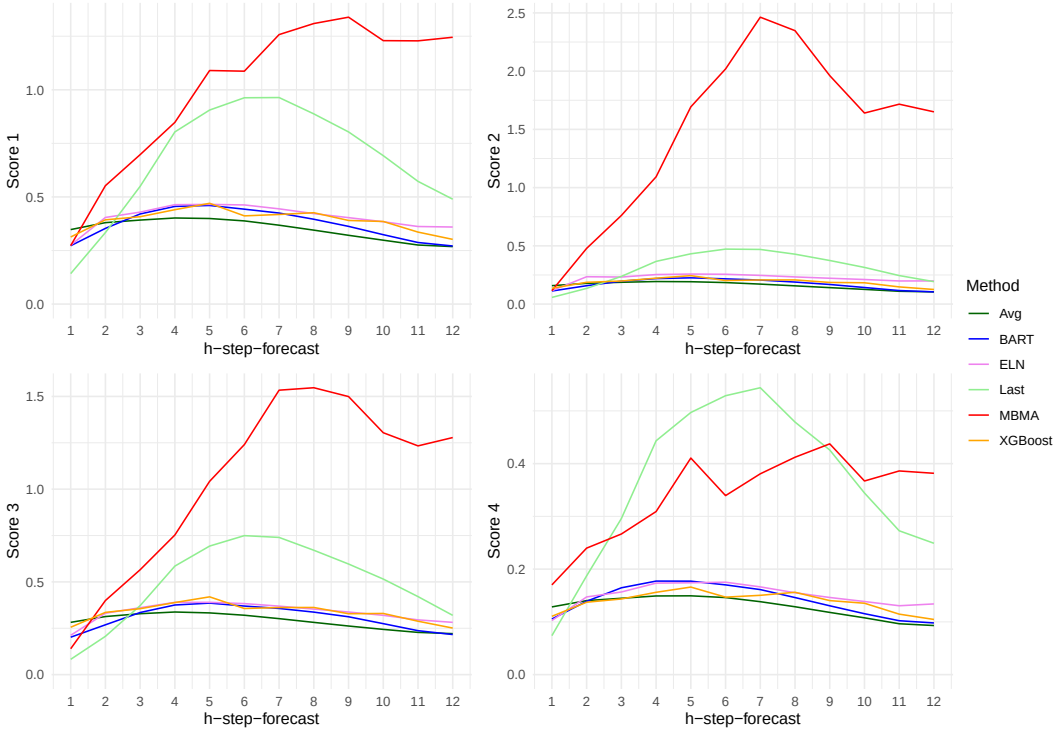


Figure 24: Compare for description of Figure 4. Sector: *Sov-HR*. Source: Deutsche Bundesbank, Bundesbank's credit register, 2008 until 2022, Own calculation

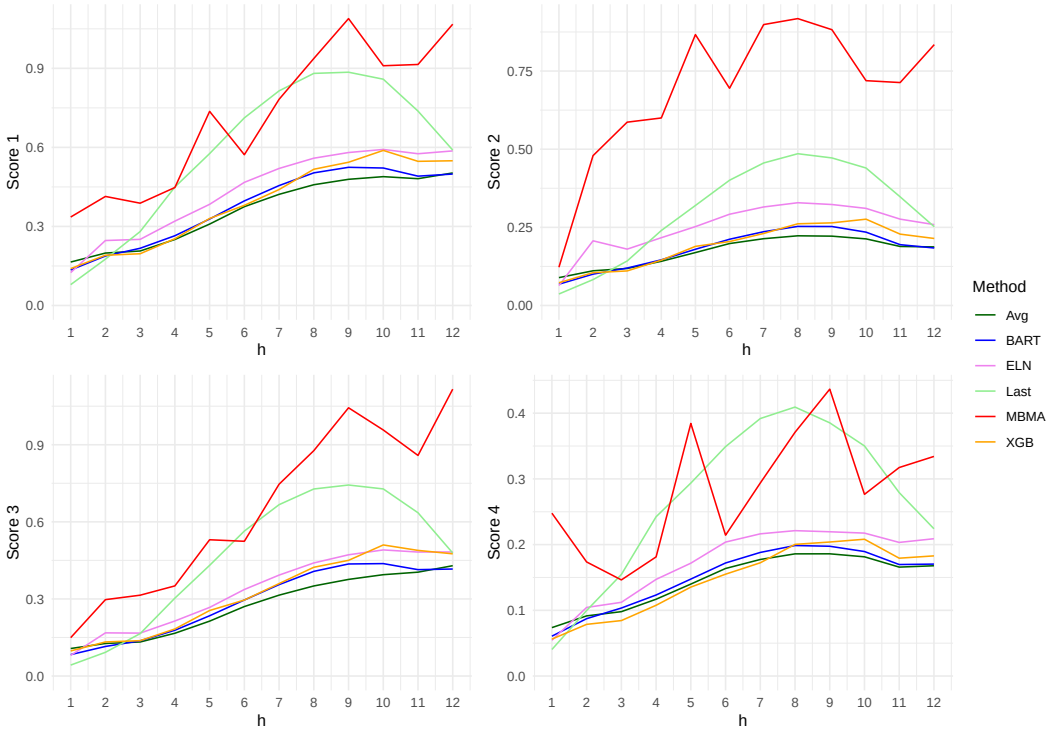


Figure 25: Compare for description of Figure 5. Sector: *Sov-HR*. Source: Deutsche Bundesbank, Bundesbank's credit register, 2008 until 2022, Own calculation

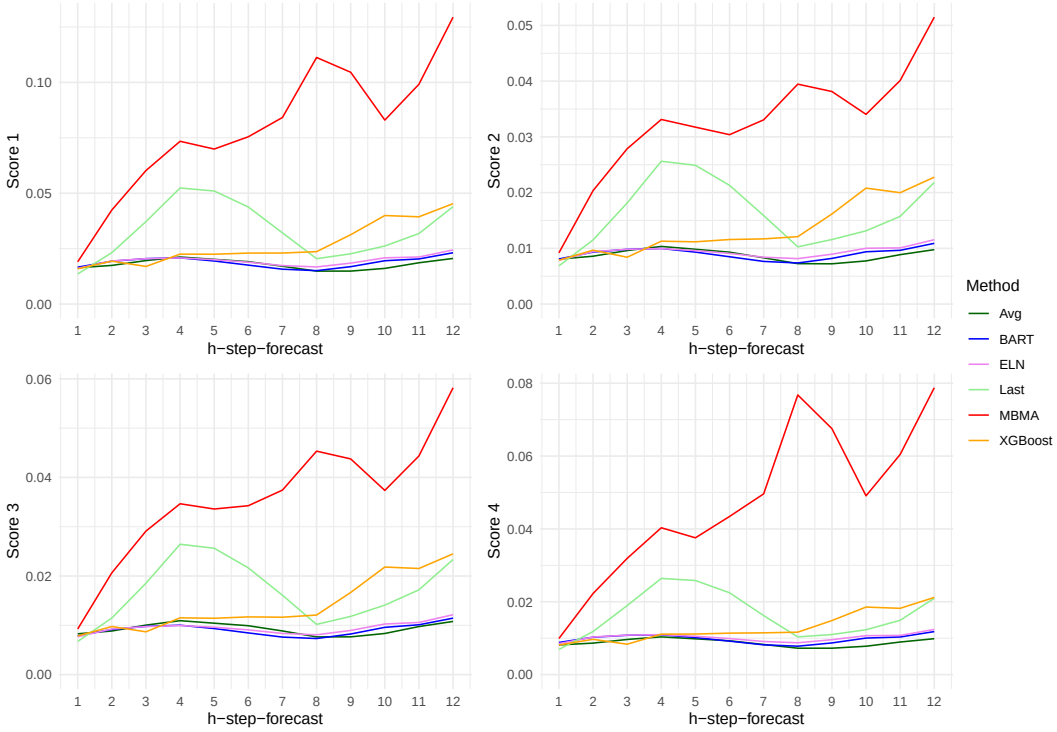


Figure 26: Compare for description of Figure 4. Sector: *HH-RE*. Source: Deutsche Bundesbank, Bundesbank's credit register, 2008 until 2022, Own calculation

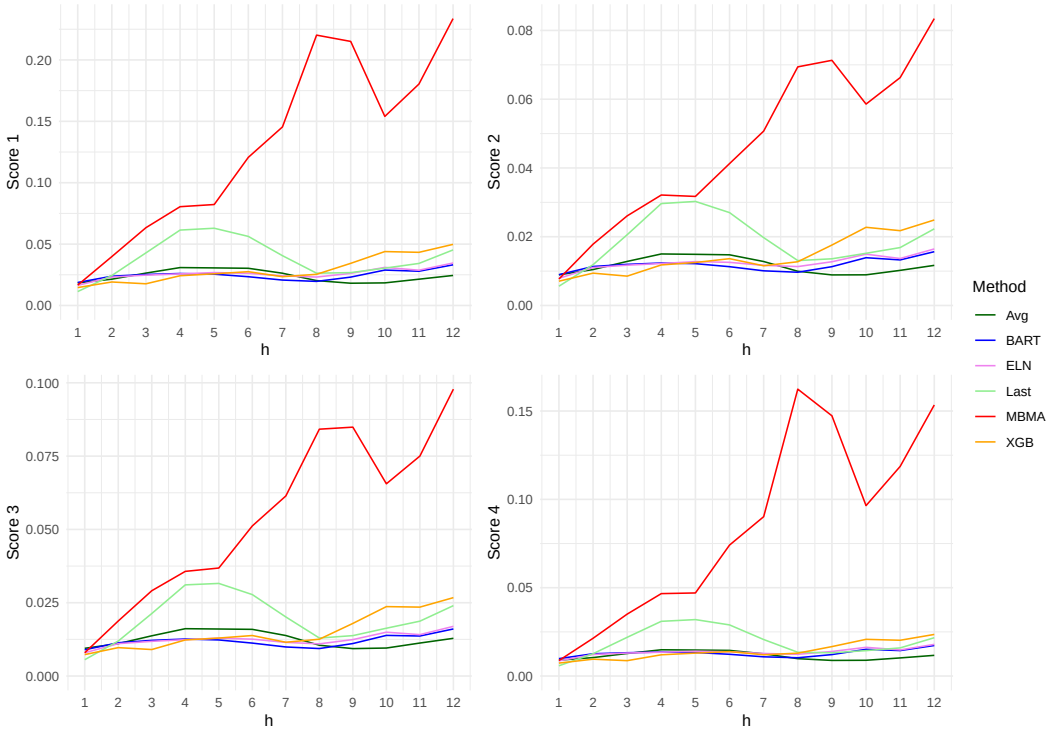


Figure 27: Compare for description of Figure 5. Sector: *HH-RE*. Source: Deutsche Bundesbank, Bundesbank's credit register, 2008 until 2022, Own calculation

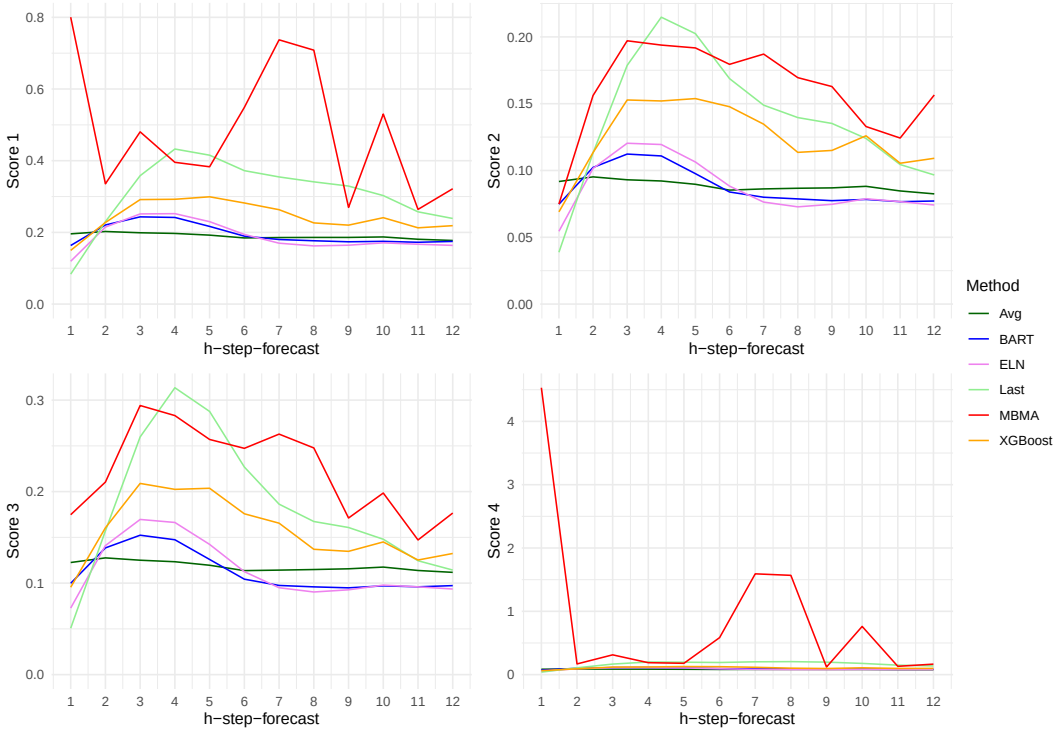


Figure 28: Compare for description of Figure 4. Sector: *NFC-RE*. Source: Deutsche Bundesbank, Bundesbank's credit register, 2008 until 2022, Own calculation

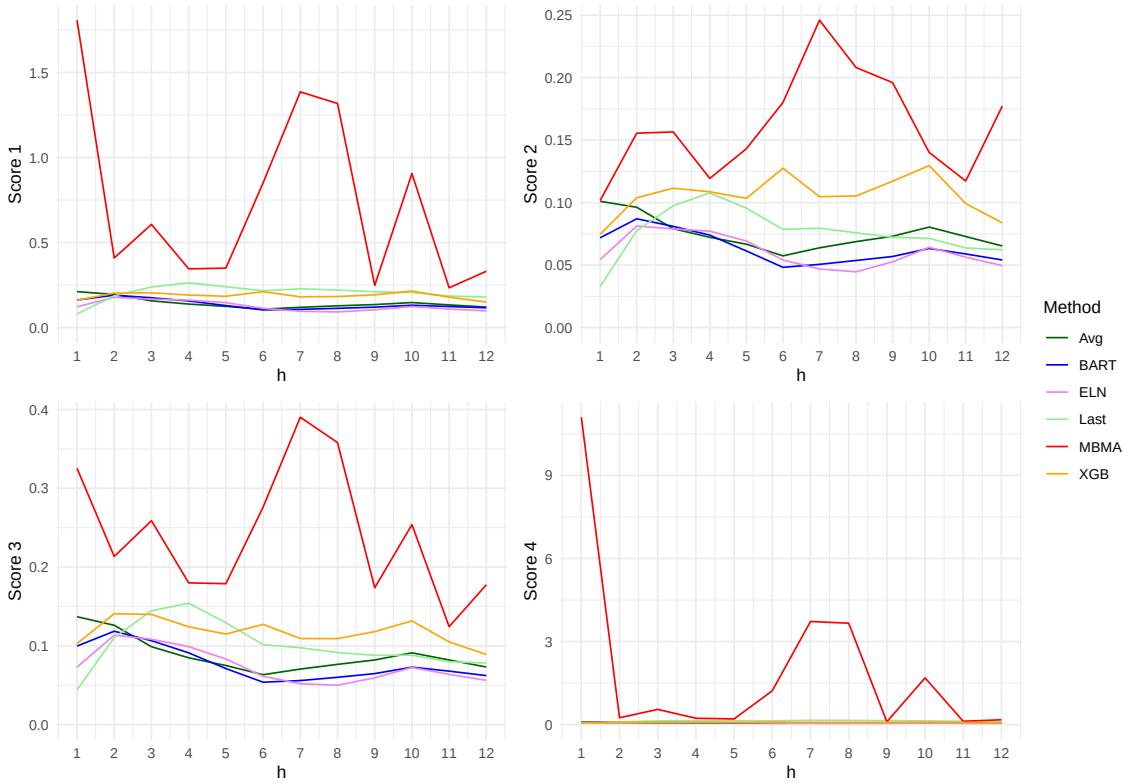


Figure 29: Compare for description of Figure 5. Sector: *NFC-RE*. Source: Deutsche Bundesbank, Bundesbank's credit register, 2008 until 2022, Own calculation

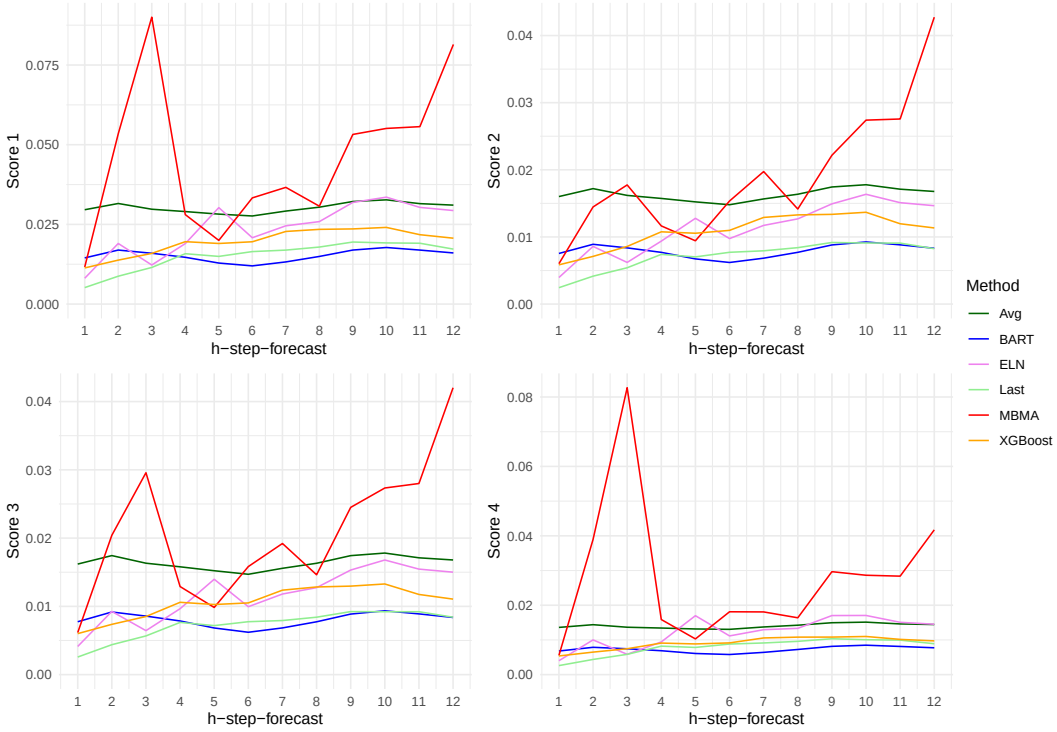


Figure 30: Compare for description of Figure 4. Sector: *NFC-nonRE*. Source: Deutsche Bundesbank, Bundesbank's credit register, 2008 until 2022, Own calculation

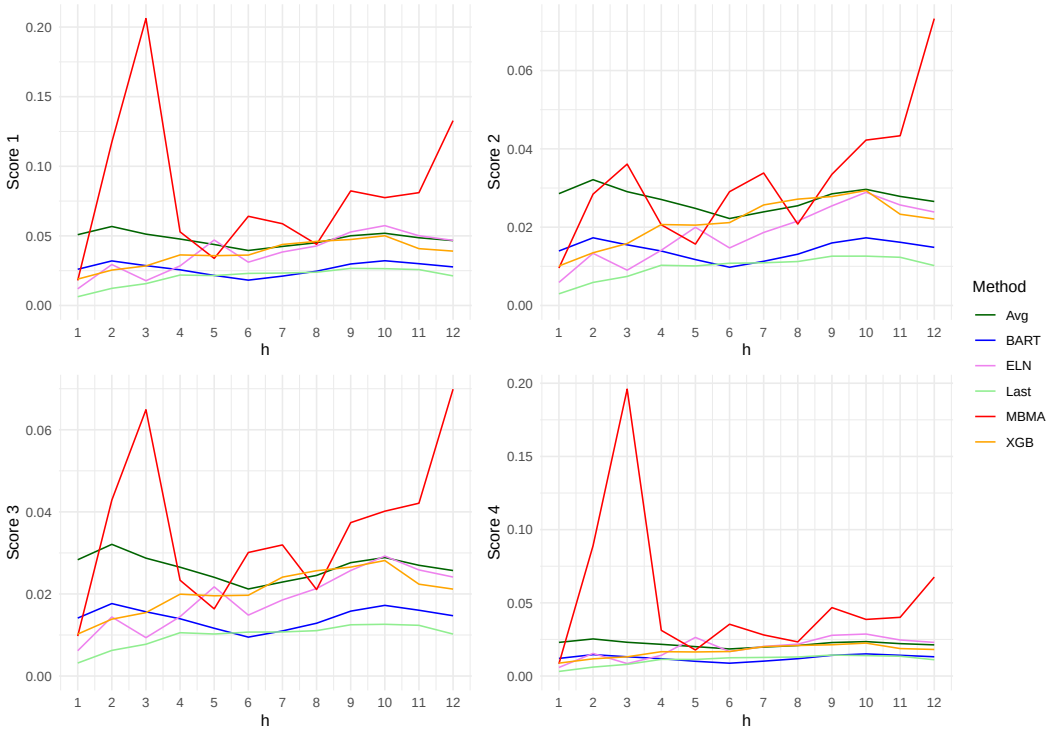


Figure 31: Compare for description of Figure 5. Sector: *NFC-nonRE*. Source: Deutsche Bundesbank, Bundesbank's credit register, 2008 until 2022, Own calculation

# C Boxplots

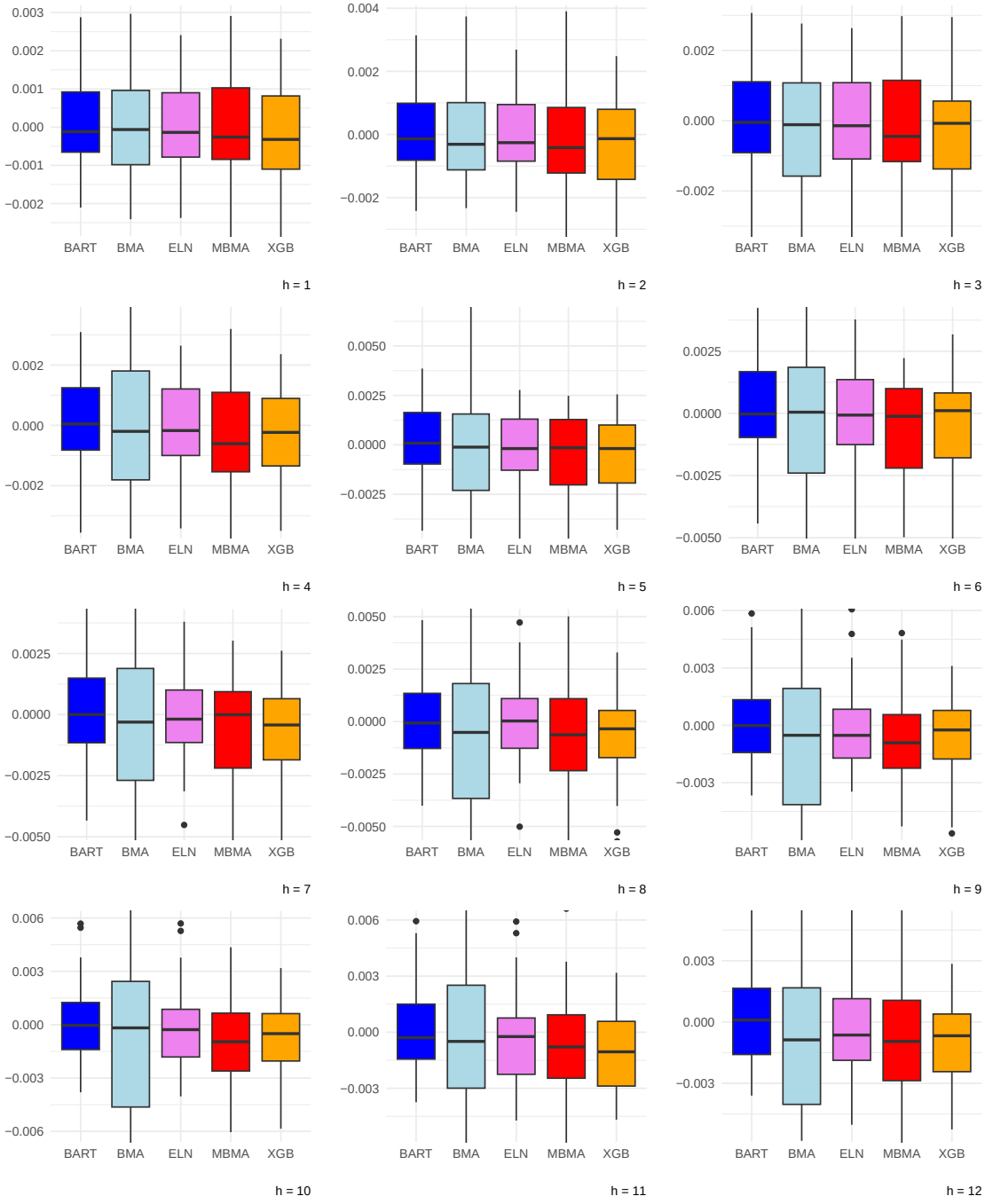


Figure 32: Compare for description of Figure 6. Sector: *HH-nonRE*. Source: Deutsche Bundesbank, Bundesbank's credit register, 2008 until 2022, Own calculation

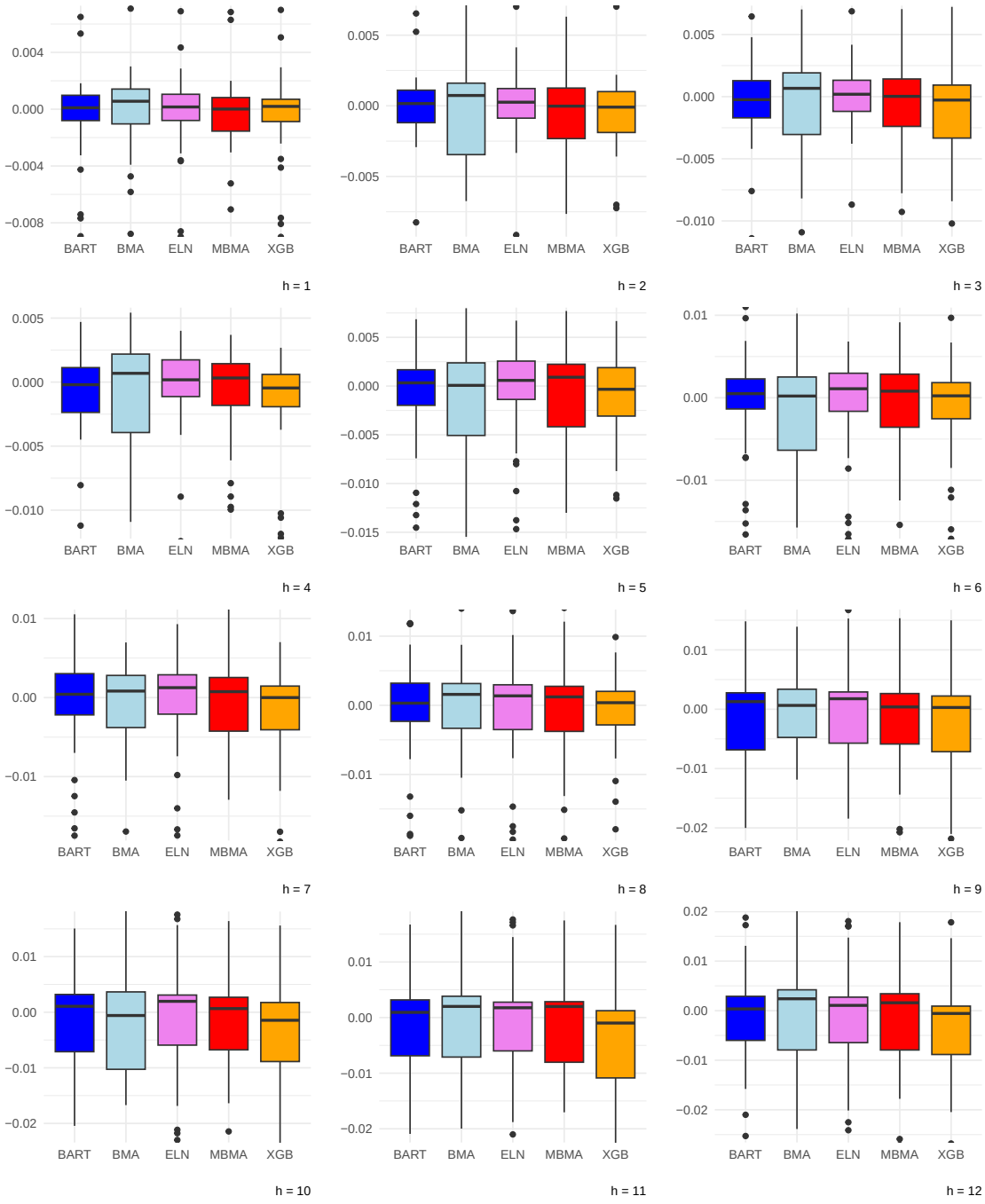


Figure 33: Compare for description of Figure 6. Sector: *Sov.* Source: Deutsche Bundesbank, Bundesbank's credit register, 2008 until 2022, Own calculation

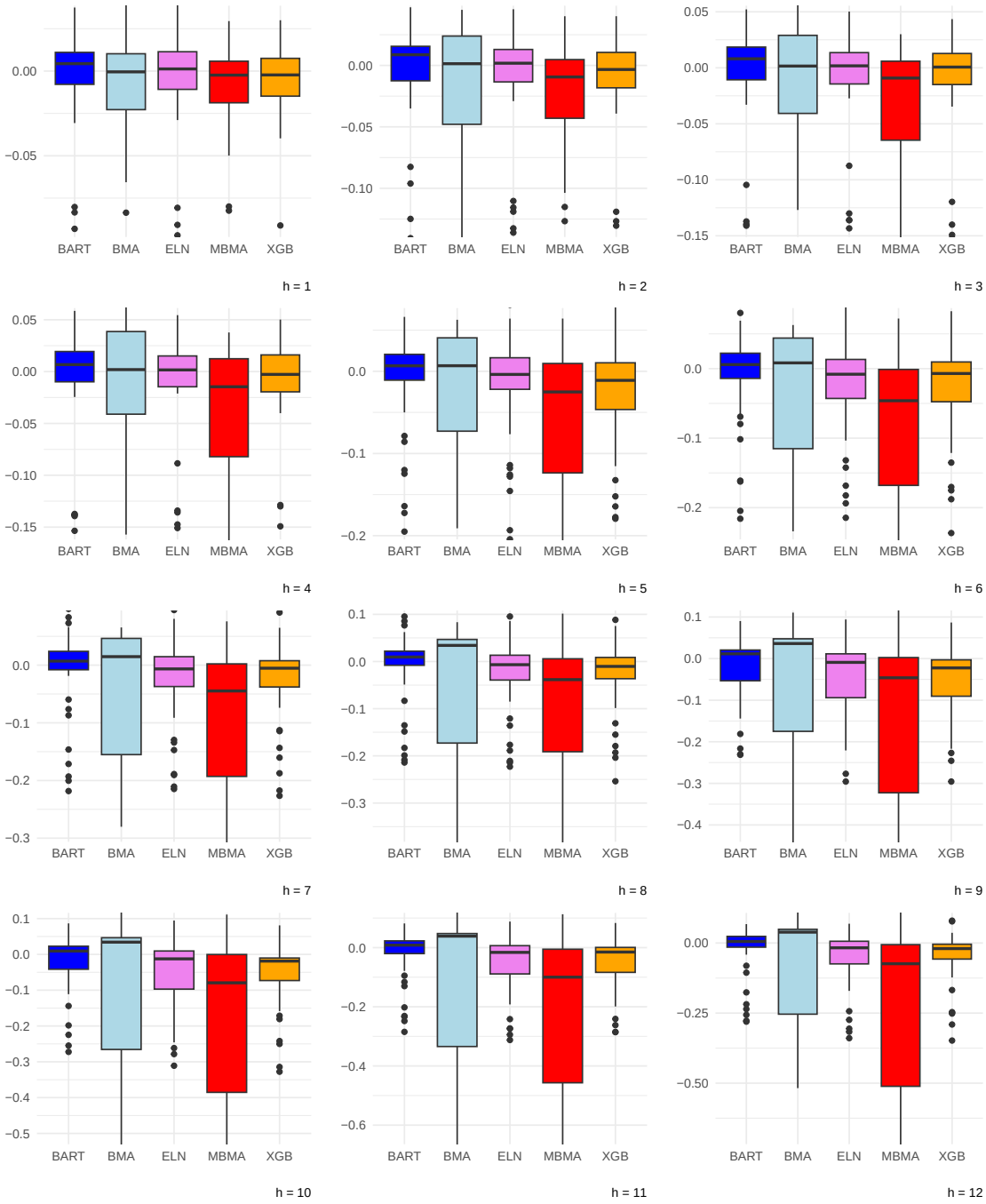


Figure 34: Compare for description of Figure 6. Sector: *Sov-HR*. Source: Deutsche Bundesbank, Bundesbank's credit register, 2008 until 2022, Own calculation

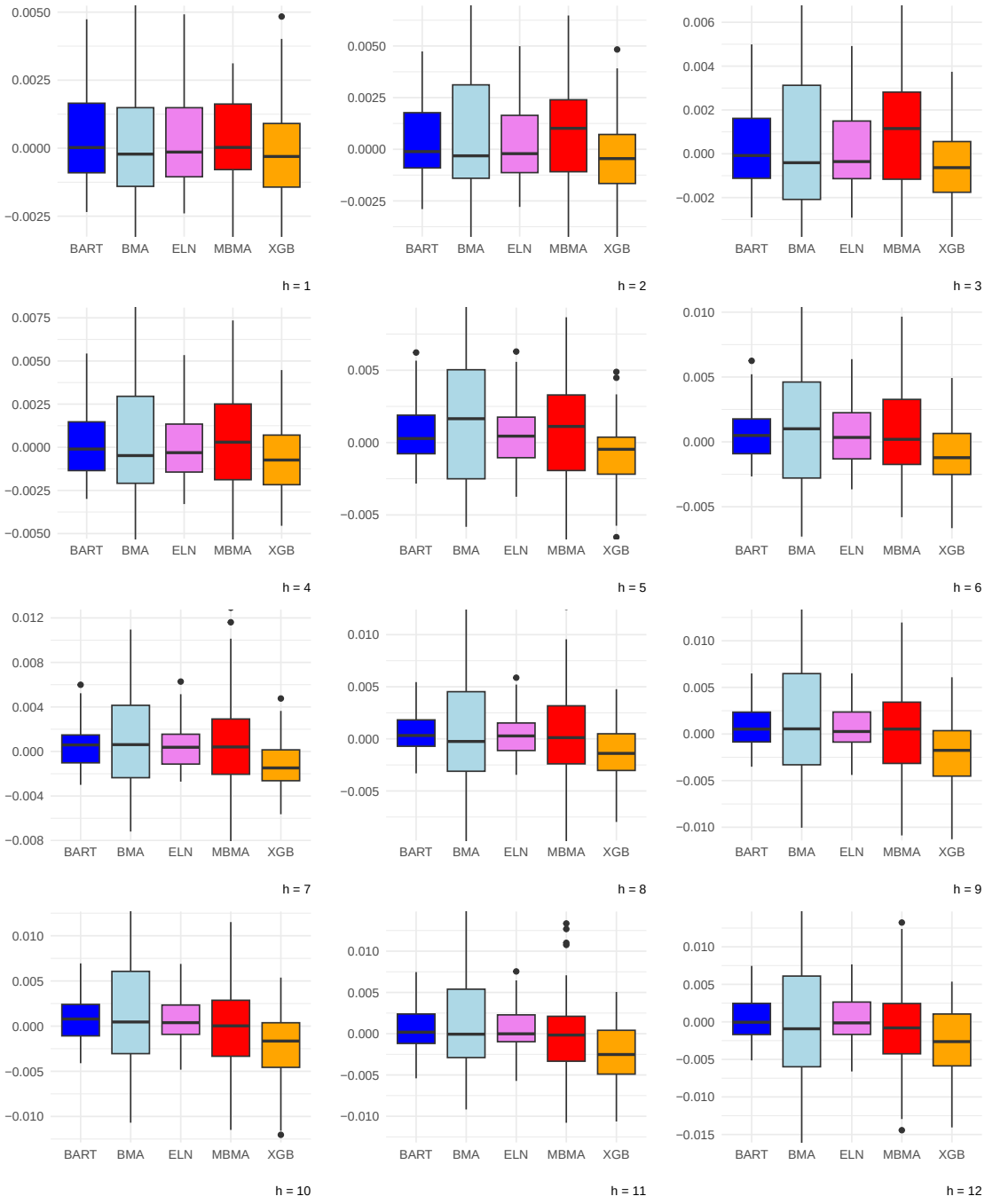


Figure 35: Compare for description of Figure 6. Sector: *HH-RE*. Source: Deutsche Bundesbank, Bundesbank's credit register, 2008 until 2022, Own calculation

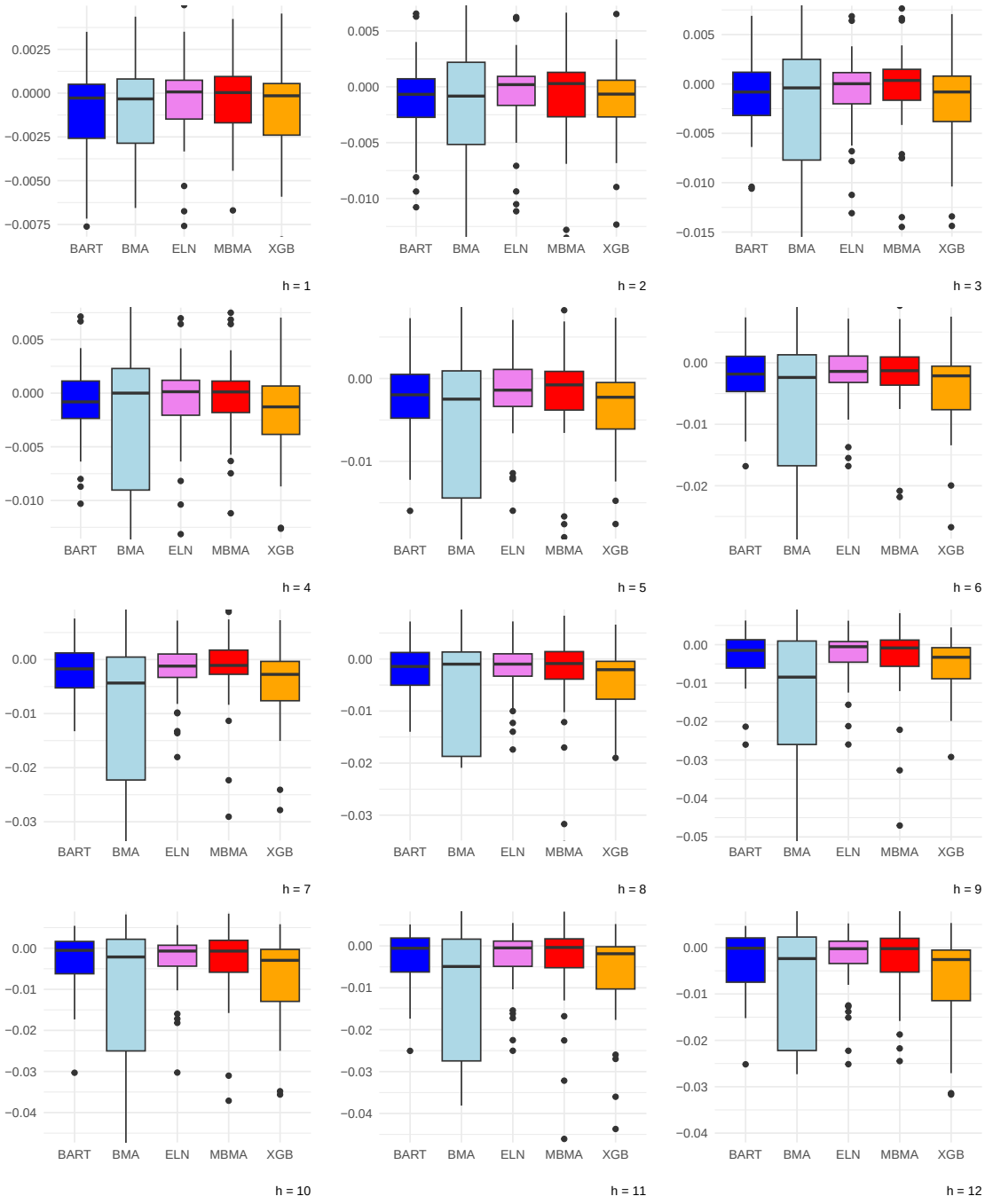


Figure 36: Compare for description of Figure 6. Sector: *NFC-RE*. Source: Deutsche Bundesbank, Bundesbank's credit register, 2008 until 2022, Own calculation

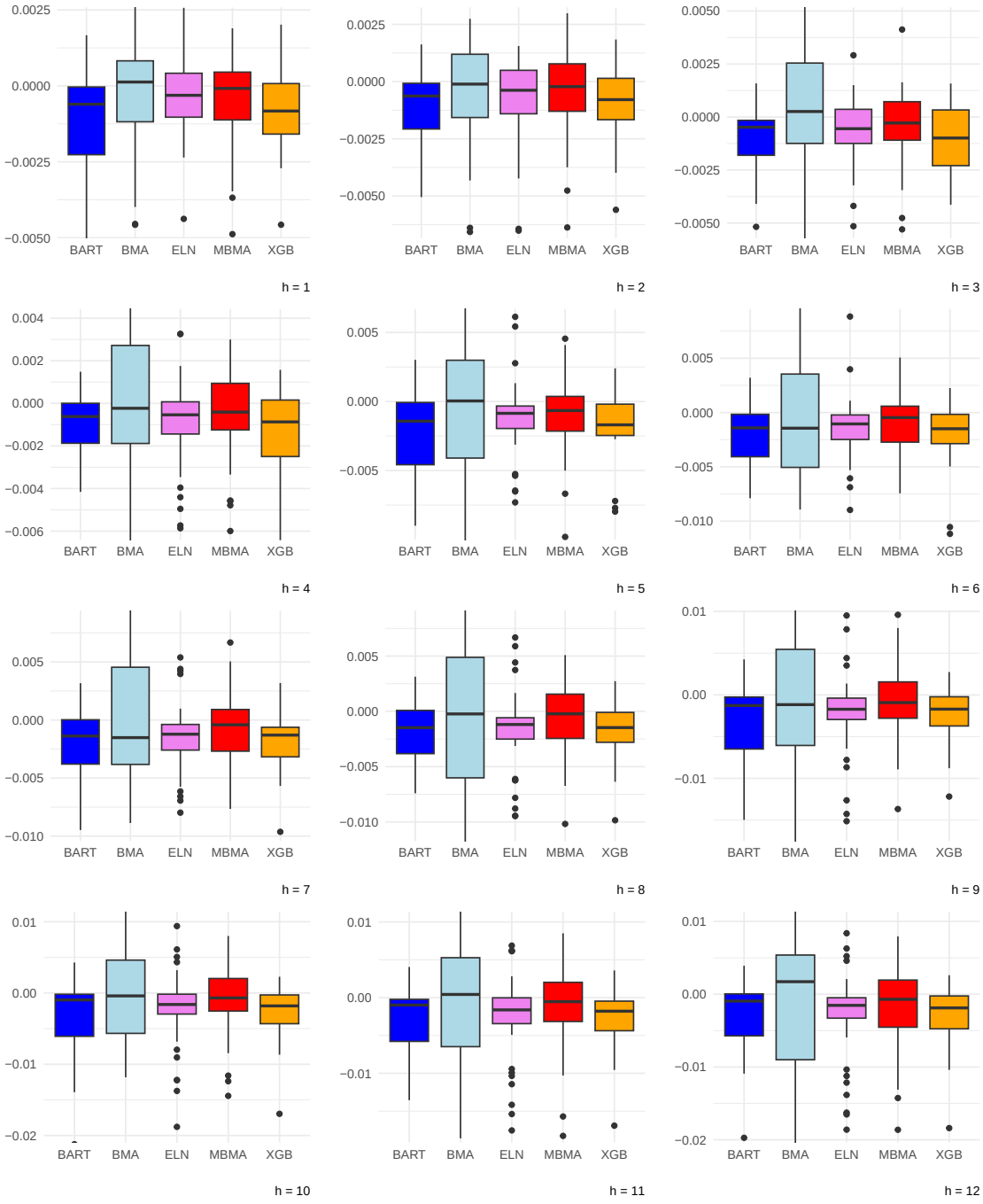


Figure 37: Compare for description of Figure 6. Sector: *NFC-nonRE*. Source: Deutsche Bundesbank, Bundesbank's credit register, 2008 until 2022, Own calculation

# D Results of the Diebold-Mariano Test

| | MBMA | BMA | BART | XGB | ELN | last | avg |
|---|---|---|---|---|---|---|---|
| MBMA | | 0.13 | 1.00 | 1.00 | 1.00 | 0.07 | 1.00 |
| BMA | 0.87 | | 0.87 | 0.87 | 0.87 | 0.87 | 0.87 |
| BART | 0.00 | 0.13 | | 0.13 | 0.02 | 0.00 | 0.90 |
| XGB | 0.00 | 0.13 | 0.87 | | 0.52 | 0.00 | 0.98 |
| ELN | 0.00 | 0.13 | 0.98 | 0.48 | | 0.00 | 0.99 |
| last | 0.93 | 0.13 | 1.00 | 1.00 | 1.00 | | 1.00 |
| avg | 0.00 | 0.13 | 0.10 | 0.02 | 0.01 | 0.00 | |

Table 16: Compare for description of table 2. Sector: *HH-nonRE*. Source: Deutsche Bundesbank, Bundesbank's credit register, 2008 until 2022, Own calculation

| | MBMA | BMA | BART | XGB | ELN | last | avg |
|---|---|---|---|---|---|---|---|
| MBMA | | 0.01 | 0.97 | 0.95 | 0.95 | 0.84 | 0.97 |
| BMA | 0.99 | | 1.00 | 1.00 | 1.00 | 0.99 | 1.00 |
| BART | 0.03 | 0.00 | | 0.00 | 0.15 | 0.01 | 0.70 |
| XGB | 0.05 | 0.00 | 1.00 | | 0.45 | 0.04 | 0.99 |
| ELN | 0.05 | 0.00 | 0.85 | 0.55 | | 0.07 | 0.86 |
| last | 0.16 | 0.01 | 0.99 | 0.96 | 0.93 | | 0.98 |
| avg | 0.03 | 0.00 | 0.30 | 0.01 | 0.14 | 0.02 | |

Table 17: Compare for description of table 2. Sector: *NFC-RE*. Source: Deutsche Bundesbank, Bundesbank's credit register, 2008 until 2022, Own calculation

| | MBMA | BMA | BART | XGB | ELN | last | avg |
|---|---|---|---|---|---|---|---|
| MBMA | | 0.16 | 0.99 | 0.99 | 0.96 | 0.99 | 0.88 |
| BMA | 0.84 | | 0.84 | 0.84 | 0.84 | 0.84 | 0.84 |
| BART | 0.01 | 0.16 | | 0.10 | 0.05 | 0.51 | 0.00 |
| XGB | 0.01 | 0.16 | 0.90 | | 0.23 | 0.72 | 0.00 |
| ELN | 0.04 | 0.16 | 0.95 | 0.77 | | 0.97 | 0.07 |
| last | 0.01 | 0.16 | 0.49 | 0.28 | 0.03 | | 0.00 |
| avg | 0.12 | 0.16 | 1.00 | 1.00 | 0.93 | 1.00 | |

Table 18: Compare for description of table 2. Sector: *NFC-nonRE*. Source: Deutsche Bundesbank, Bundesbank's credit register, 2008 until 2022, Own calculation

| | MBMA | BMA | BART | XGB | ELN | last | avg |
|---|---|---|---|---|---|---|---|
| MBMA | | 0.12 | 1.00 | 1.00 | 1.00 | 0.99 | 1.00 |
| BMA | 0.88 | | 0.88 | 0.88 | 0.88 | 0.88 | 0.88 |
| BART | 0.00 | 0.12 | | 0.00 | 0.01 | 0.00 | 0.80 |
| XGB | 0.00 | 0.12 | 1.00 | | 1.00 | 0.01 | 1.00 |
| ELN | 0.00 | 0.12 | 0.99 | 0.00 | | 0.00 | 0.96 |
| last | 0.01 | 0.12 | 1.00 | 0.99 | 1.00 | | 1.00 |
| avg | 0.00 | 0.12 | 0.20 | 0.00 | 0.04 | 0.00 | |

Table 19: Compare for description of table 2. Sector: *HH-RE*. Source: Deutsche Bundesbank, Bundesbank's credit register, 2008 until 2022, Own calculation

| | MBMA | BMA | BART | XGB | ELN | last | avg |
|---|---|---|---|---|---|---|---|
| MBMA | | 0.16 | 1.00 | 1.00 | 1.00 | 0.97 | 1.00 |
| BMA | 0.84 | | 0.84 | 0.84 | 0.84 | 0.84 | 0.84 |
| BART | 0.00 | 0.16 | | 0.02 | 0.01 | 0.00 | 0.99 |
| XGB | 0.00 | 0.16 | 0.98 | | 0.24 | 0.00 | 1.00 |
| ELN | 0.00 | 0.16 | 0.99 | 0.76 | | 0.00 | 1.00 |
| last | 0.03 | 0.16 | 1.00 | 1.00 | 1.00 | | 1.00 |
| avg | 0.00 | 0.16 | 0.01 | 0.00 | 0.00 | 0.00 | |

Table 20: Compare for description of table 2. Sector: *Sov-HR*. Source: Deutsche Bundesbank, Bundesbank's credit register, 2008 until 2022, Own calculation

| | MBMA | BMA | BART | XGB | ELN | last | avg |
|---|---|---|---|---|---|---|---|
| MBMA | | 0.00 | 1.00 | 1.00 | 1.00 | 0.38 | 1.00 |
| BMA | 1.00 | | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| BART | 0.00 | 0.00 | | 0.99 | 0.20 | 0.01 | 0.74 |
| XGB | 0.00 | 0.00 | 0.01 | | 0.00 | 0.00 | 0.00 |
| ELN | 0.00 | 0.00 | 0.80 | 1.00 | | 0.02 | 1.00 |
| last | 0.62 | 0.00 | 0.99 | 1.00 | 0.98 | | 0.99 |
| avg | 0.00 | 0.00 | 0.26 | 1.00 | 0.00 | 0.01 | |

Table 21: Compare for description of table 2. Sector: *Sov*. Source: Deutsche Bundesbank, Bundesbank's credit register, 2008 until 2022, Own calculation

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.02 | 0.82 | 0.63 | 0.82 |
| BMA | 0.98 | | 0.98 | 0.98 | 0.98 |
| BART | 0.18 | 0.02 | | 0.18 | 0.34 |
| XGB | 0.37 | 0.02 | 0.82 | | 0.78 |
| ELN | 0.18 | 0.02 | 0.66 | 0.22 | |

Table 22: 1-step forecast

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.10 | 0.93 | 0.69 | 0.93 |
| BMA | 0.90 | | 0.90 | 0.89 | 0.90 |
| BART | 0.07 | 0.10 | | 0.06 | 0.20 |
| XGB | 0.31 | 0.11 | 0.94 | | 0.93 |
| ELN | 0.07 | 0.10 | 0.80 | 0.07 | |

Table 23: 2-step forecast

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.13 | 0.92 | 0.64 | 0.89 |
| BMA | 0.87 | | 0.87 | 0.87 | 0.87 |
| BART | 0.08 | 0.13 | | 0.11 | 0.21 |
| XGB | 0.36 | 0.13 | 0.89 | | 0.85 |
| ELN | 0.11 | 0.13 | 0.79 | 0.15 | |

Table 24: 3-step forecast

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.13 | 0.92 | 0.79 | 0.93 |
| BMA | 0.87 | | 0.87 | 0.87 | 0.87 |
| BART | 0.08 | 0.13 | | 0.17 | 0.46 |
| XGB | 0.21 | 0.13 | 0.83 | | 0.81 |
| ELN | 0.07 | 0.13 | 0.54 | 0.19 | |

Table 25: 4-step forecast

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.14 | 0.88 | 0.87 | 0.87 |
| BMA | 0.86 | | 0.86 | 0.86 | 0.86 |
| BART | 0.12 | 0.14 | | 0.68 | 0.37 |
| XGB | 0.13 | 0.14 | 0.32 | | 0.32 |
| ELN | 0.13 | 0.14 | 0.63 | 0.68 | |

Table 26: 5-step forecast

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.14 | 0.82 | 0.77 | 0.84 |
| BMA | 0.86 | | 0.86 | 0.86 | 0.86 |
| BART | 0.18 | 0.14 | | 0.65 | 0.32 |
| XGB | 0.23 | 0.14 | 0.35 | | 0.33 |
| ELN | 0.16 | 0.14 | 0.68 | 0.67 | |

Table 27: 6-step forecast

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.15 | 0.83 | 0.78 | 0.95 |
| BMA | 0.85 | | 0.85 | 0.85 | 0.85 |
| BART | 0.17 | 0.15 | | 0.58 | 0.42 |
| XGB | 0.22 | 0.15 | 0.42 | | 0.40 |
| ELN | 0.05 | 0.15 | 0.58 | 0.60 | |

Table 28: 7-step forecast

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.14 | 0.99 | 0.89 | 1.00 |
| BMA | 0.86 | | 0.86 | 0.86 | 0.86 |
| BART | 0.01 | 0.14 | | 0.43 | 0.29 |
| XGB | 0.11 | 0.14 | 0.57 | | 0.51 |
| ELN | 0.00 | 0.14 | 0.71 | 0.49 | |

Table 29: 8-step forecast

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.14 | 0.89 | 0.86 | 0.85 |
| BMA | 0.86 | | 0.86 | 0.86 | 0.86 |
| BART | 0.11 | 0.14 | | 0.32 | 0.04 |
| XGB | 0.14 | 0.14 | 0.68 | | 0.58 |
| ELN | 0.15 | 0.14 | 0.96 | 0.42 | |

Table 30: 9-step forecast

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.13 | 0.93 | 0.98 | 0.88 |
| BMA | 0.87 | | 0.87 | 0.87 | 0.87 |
| BART | 0.07 | 0.13 | | 0.34 | 0.05 |
| XGB | 0.02 | 0.13 | 0.66 | | 0.51 |
| ELN | 0.12 | 0.13 | 0.95 | 0.49 | |

Table 31: 10-step forecast

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.12 | 0.95 | 1.00 | 0.92 |
| BMA | 0.88 | | 0.88 | 0.88 | 0.88 |
| BART | 0.05 | 0.12 | | 0.54 | 0.05 |
| XGB | 0.00 | 0.12 | 0.46 | | 0.20 |
| ELN | 0.08 | 0.12 | 0.95 | 0.80 | |

Table 32: 11-step forecast

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.11 | 0.91 | 0.92 | 0.88 |
| BMA | 0.89 | | 0.89 | 0.89 | 0.89 |
| BART | 0.09 | 0.11 | | 0.43 | 0.09 |
| XGB | 0.08 | 0.11 | 0.57 | | 0.08 |
| ELN | 0.12 | 0.11 | 0.91 | 0.92 | |

Table 33: 12-step forecast

Table 34: Compare for description of table 15. Sector: *HH-nonRE*. Source: Deutsche Bundesbank, Bundesbank's credit register, 2008 until 2022, Own calculation

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.02 | 0.80 | 0.71 | 0.63 |
| BMA | 0.98 | | 0.98 | 0.97 | 0.97 |
| BART | 0.20 | 0.02 | | 0.11 | 0.02 |
| XGB | 0.29 | 0.03 | 0.89 | | 0.12 |
| ELN | 0.37 | 0.03 | 0.98 | 0.88 | |

Table 35: 1-step forecast

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.01 | 0.85 | 0.81 | 0.79 |
| BMA | 0.99 | | 0.97 | 0.97 | 0.97 |
| BART | 0.15 | 0.03 | | 0.27 | 0.16 |
| XGB | 0.19 | 0.03 | 0.73 | | 0.35 |
| ELN | 0.21 | 0.03 | 0.84 | 0.65 | |

Table 36: 2-step forecast

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.03 | 0.84 | 0.83 | 0.82 |
| BMA | 0.97 | | 0.96 | 0.95 | 0.95 |
| BART | 0.16 | 0.04 | | 0.43 | 0.45 |
| XGB | 0.17 | 0.05 | 0.57 | | 0.50 |
| ELN | 0.18 | 0.05 | 0.55 | 0.50 | |

Table 37: 3-step forecast

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.05 | 0.88 | 0.91 | 0.90 |
| BMA | 0.95 | | 0.94 | 0.94 | 0.94 |
| BART | 0.12 | 0.06 | | 0.58 | 0.63 |
| XGB | 0.09 | 0.06 | 0.42 | | 0.54 |
| ELN | 0.10 | 0.06 | 0.37 | 0.46 | |

Table 38: 4-step forecast

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.07 | 0.85 | 0.92 | 0.85 |
| BMA | 0.93 | | 0.92 | 0.93 | 0.92 |
| BART | 0.15 | 0.08 | | 0.88 | 0.59 |
| XGB | 0.08 | 0.07 | 0.12 | | 0.12 |
| ELN | 0.15 | 0.08 | 0.41 | 0.88 | |

Table 39: 5-step forecast

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.04 | 0.82 | 0.87 | 0.81 |
| BMA | 0.96 | | 0.94 | 0.95 | 0.94 |
| BART | 0.18 | 0.06 | | 0.71 | 0.61 |
| XGB | 0.13 | 0.05 | 0.29 | | 0.32 |
| ELN | 0.19 | 0.06 | 0.39 | 0.68 | |

Table 40: 6-step forecast

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.11 | 0.91 | 0.97 | 0.93 |
| BMA | 0.89 | | 0.92 | 0.94 | 0.92 |
| BART | 0.09 | 0.08 | | 0.99 | 0.54 |
| XGB | 0.03 | 0.06 | 0.01 | | 0.01 |
| ELN | 0.07 | 0.08 | 0.46 | 0.99 | |

Table 41: 7-step forecast

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.04 | 0.84 | 0.92 | 0.86 |
| BMA | 0.96 | | 0.99 | 0.99 | 0.99 |
| BART | 0.16 | 0.01 | | 1.00 | 0.74 |
| XGB | 0.08 | 0.01 | 0.00 | | 0.01 |
| ELN | 0.14 | 0.01 | 0.26 | 0.99 | |

Table 42: 8-step forecast

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.02 | 0.67 | 0.80 | 0.67 |
| BMA | 0.98 | | 0.98 | 0.98 | 0.98 |
| BART | 0.33 | 0.02 | | 0.99 | 0.32 |
| XGB | 0.20 | 0.02 | 0.01 | | 0.02 |
| ELN | 0.33 | 0.02 | 0.68 | 0.98 | |

Table 43: 9-step forecast

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.06 | 0.78 | 0.85 | 0.77 |
| BMA | 0.94 | | 0.93 | 0.94 | 0.94 |
| BART | 0.22 | 0.07 | | 0.98 | 0.18 |
| XGB | 0.15 | 0.06 | 0.02 | | 0.07 |
| ELN | 0.23 | 0.06 | 0.82 | 0.93 | |

Table 44: 10-step forecast

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.09 | 0.80 | 0.84 | 0.77 |
| BMA | 0.91 | | 0.91 | 0.92 | 0.91 |
| BART | 0.20 | 0.09 | | 0.94 | 0.16 |
| XGB | 0.16 | 0.08 | 0.06 | | 0.09 |
| ELN | 0.23 | 0.09 | 0.84 | 0.91 | |

Table 45: 11-step forecast

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.08 | 0.93 | 0.85 | 0.98 |
| BMA | 0.92 | | 0.92 | 0.92 | 0.92 |
| BART | 0.07 | 0.08 | | 0.82 | 0.25 |
| XGB | 0.15 | 0.08 | 0.18 | | 0.17 |
| ELN | 0.02 | 0.08 | 0.75 | 0.83 | |

Table 46: 12-step forecast

Table 47: Compare for description of table 15. Sector: *Sov.* Source: Deutsche Bundesbank, Bundesbank's credit register, 2008 until 2022, Own calculation

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.04 | 0.49 | 0.36 | 0.49 |
| BMA | 0.96 | | 0.96 | 0.95 | 0.96 |
| BART | 0.51 | 0.04 | | 0.08 | 0.52 |
| XGB | 0.64 | 0.05 | 0.92 | | 0.94 |
| ELN | 0.51 | 0.04 | 0.48 | 0.06 | |

Table 48: 1-step forecast

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.02 | 0.89 | 0.83 | 0.83 |
| BMA | 0.98 | | 0.99 | 0.99 | 0.99 |
| BART | 0.11 | 0.01 | | 0.28 | 0.18 |
| XGB | 0.17 | 0.01 | 0.72 | | 0.34 |
| ELN | 0.17 | 0.01 | 0.82 | 0.66 | |

Table 49: 2-step forecast

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.06 | 0.86 | 0.83 | 0.85 |
| BMA | 0.94 | | 0.94 | 0.94 | 0.94 |
| BART | 0.14 | 0.06 | | 0.33 | 0.37 |
| XGB | 0.17 | 0.06 | 0.67 | | 0.60 |
| ELN | 0.15 | 0.06 | 0.63 | 0.40 | |

Table 50: 3-step forecast

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.14 | 0.84 | 0.84 | 0.84 |
| BMA | 0.86 | | 0.87 | 0.87 | 0.87 |
| BART | 0.16 | 0.13 | | 0.56 | 0.43 |
| XGB | 0.16 | 0.13 | 0.44 | | 0.38 |
| ELN | 0.16 | 0.13 | 0.57 | 0.62 | |

Table 51: 4-step forecast

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.18 | 0.89 | 0.90 | 0.88 |
| BMA | 0.82 | | 0.82 | 0.82 | 0.82 |
| BART | 0.11 | 0.18 | | 0.74 | 0.51 |
| XGB | 0.10 | 0.18 | 0.26 | | 0.34 |
| ELN | 0.12 | 0.18 | 0.49 | 0.66 | |

Table 52: 5-step forecast

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.19 | 0.84 | 0.84 | 0.82 |
| BMA | 0.81 | | 0.81 | 0.81 | 0.81 |
| BART | 0.16 | 0.19 | | 0.88 | 0.44 |
| XGB | 0.16 | 0.19 | 0.12 | | 0.29 |
| ELN | 0.18 | 0.19 | 0.56 | 0.71 | |

Table 53: 6-step forecast

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.20 | 0.88 | 0.88 | 0.87 |
| BMA | 0.80 | | 0.80 | 0.80 | 0.80 |
| BART | 0.12 | 0.20 | | 0.50 | 0.40 |
| XGB | 0.12 | 0.20 | 0.50 | | 0.39 |
| ELN | 0.13 | 0.20 | 0.60 | 0.61 | |

Table 54: 7-step forecast

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.20 | 0.93 | 0.92 | 0.90 |
| BMA | 0.80 | | 0.80 | 0.80 | 0.80 |
| BART | 0.07 | 0.20 | | 0.18 | 0.35 |
| XGB | 0.08 | 0.20 | 0.82 | | 0.47 |
| ELN | 0.10 | 0.20 | 0.65 | 0.53 | |

Table 55: 8-step forecast

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.21 | 0.96 | 0.96 | 0.93 |
| BMA | 0.79 | | 0.79 | 0.79 | 0.79 |
| BART | 0.04 | 0.21 | | 0.14 | 0.21 |
| XGB | 0.04 | 0.21 | 0.86 | | 0.39 |
| ELN | 0.07 | 0.21 | 0.79 | 0.61 | |

Table 56: 9-step forecast

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.22 | 0.93 | 0.92 | 0.90 |
| BMA | 0.78 | | 0.78 | 0.78 | 0.78 |
| BART | 0.07 | 0.22 | | 0.04 | 0.04 |
| XGB | 0.08 | 0.22 | 0.96 | | 0.47 |
| ELN | 0.10 | 0.22 | 0.96 | 0.53 | |

Table 57: 10-step forecast

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.22 | 0.92 | 0.91 | 0.89 |
| BMA | 0.78 | | 0.78 | 0.78 | 0.78 |
| BART | 0.08 | 0.22 | | 0.11 | 0.00 |
| XGB | 0.09 | 0.22 | 0.89 | | 0.41 |
| ELN | 0.11 | 0.22 | 1.00 | 0.59 | |

Table 58: 11-step forecast

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.23 | 0.94 | 0.94 | 0.92 |
| BMA | 0.77 | | 0.77 | 0.77 | 0.77 |
| BART | 0.06 | 0.23 | | 0.13 | 0.00 |
| XGB | 0.06 | 0.23 | 0.87 | | 0.37 |
| ELN | 0.08 | 0.23 | 1.00 | 0.63 | |

Table 59: 12-step forecast

Table 60: Compare for description of table 15. Sector: *Sov-HR*. Source: Deutsche Bundesbank, Bundesbank's credit register, 2008 until 2022, Own calculation

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.12 | 0.70 | 0.68 | 0.72 |
| BMA | 0.88 | | 0.88 | 0.88 | 0.88 |
| BART | 0.30 | 0.12 | | 0.52 | 0.57 |
| XGB | 0.32 | 0.12 | 0.48 | | 0.50 |
| ELN | 0.28 | 0.12 | 0.43 | 0.50 | |

Table 61: 1-step forecast

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.15 | 0.97 | 0.96 | 0.97 |
| BMA | 0.85 | | 0.86 | 0.86 | 0.86 |
| BART | 0.03 | 0.14 | | 0.57 | 0.35 |
| XGB | 0.04 | 0.14 | 0.43 | | 0.40 |
| ELN | 0.03 | 0.14 | 0.65 | 0.60 | |

Table 62: 2-step forecast

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.16 | 1.00 | 0.99 | 1.00 |
| BMA | 0.84 | | 0.84 | 0.84 | 0.84 |
| BART | 0.00 | 0.16 | | 0.56 | 0.63 |
| XGB | 0.01 | 0.16 | 0.44 | | 0.45 |
| ELN | 0.00 | 0.16 | 0.37 | 0.55 | |

Table 63: 3-step forecast

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.16 | 0.99 | 0.97 | 0.99 |
| BMA | 0.84 | | 0.84 | 0.84 | 0.84 |
| BART | 0.01 | 0.16 | | 0.32 | 0.42 |
| XGB | 0.03 | 0.16 | 0.68 | | 0.67 |
| ELN | 0.01 | 0.16 | 0.58 | 0.33 | |

Table 64: 4-step forecast

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.16 | 0.98 | 0.96 | 0.98 |
| BMA | 0.84 | | 0.84 | 0.84 | 0.84 |
| BART | 0.02 | 0.16 | | 0.38 | 0.33 |
| XGB | 0.04 | 0.16 | 0.62 | | 0.56 |
| ELN | 0.02 | 0.16 | 0.67 | 0.44 | |

Table 65: 5-step forecast

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.16 | 0.92 | 0.87 | 0.92 |
| BMA | 0.84 | | 0.84 | 0.84 | 0.84 |
| BART | 0.08 | 0.16 | | 0.13 | 0.21 |
| XGB | 0.13 | 0.16 | 0.87 | | 0.81 |
| ELN | 0.08 | 0.16 | 0.79 | 0.19 | |

Table 66: 6-step forecast

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.15 | 0.89 | 0.86 | 0.90 |
| BMA | 0.85 | | 0.85 | 0.85 | 0.85 |
| BART | 0.11 | 0.15 | | 0.10 | 0.13 |
| XGB | 0.14 | 0.15 | 0.90 | | 0.79 |
| ELN | 0.10 | 0.15 | 0.87 | 0.21 | |

Table 67: 7-step forecast

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.15 | 0.87 | 0.84 | 0.87 |
| BMA | 0.85 | | 0.85 | 0.85 | 0.85 |
| BART | 0.13 | 0.15 | | 0.06 | 0.10 |
| XGB | 0.16 | 0.15 | 0.94 | | 0.86 |
| ELN | 0.13 | 0.15 | 0.90 | 0.14 | |

Table 68: 8-step forecast

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.14 | 0.86 | 0.82 | 0.86 |
| BMA | 0.86 | | 0.86 | 0.86 | 0.86 |
| BART | 0.14 | 0.14 | | 0.10 | 0.14 |
| XGB | 0.18 | 0.14 | 0.90 | | 0.81 |
| ELN | 0.14 | 0.14 | 0.86 | 0.19 | |

Table 69: 9-step forecast

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.13 | 0.90 | 0.81 | 0.89 |
| BMA | 0.87 | | 0.87 | 0.87 | 0.87 |
| BART | 0.10 | 0.13 | | 0.08 | 0.05 |
| XGB | 0.19 | 0.13 | 0.92 | | 0.89 |
| ELN | 0.11 | 0.13 | 0.95 | 0.11 | |

Table 70: 10-step forecast

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.12 | 0.91 | 0.82 | 0.90 |
| BMA | 0.88 | | 0.88 | 0.88 | 0.88 |
| BART | 0.09 | 0.12 | | 0.11 | 0.08 |
| XGB | 0.18 | 0.12 | 0.89 | | 0.88 |
| ELN | 0.10 | 0.12 | 0.92 | 0.12 | |

Table 71: 11-step forecast

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.11 | 0.88 | 0.81 | 0.88 |
| BMA | 0.89 | | 0.89 | 0.89 | 0.89 |
| BART | 0.12 | 0.11 | | 0.06 | 0.20 |
| XGB | 0.19 | 0.11 | 0.94 | | 0.93 |
| ELN | 0.12 | 0.11 | 0.80 | 0.07 | |

Table 72: 12-step forecast

Table 73: Compare for description of table 15. Sector: *HH-RE*. Source: Deutsche Bundesbank, Bundesbank's credit register, 2008 until 2022, Own calculation

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.77 | 0.82 | 0.82 | 0.83 |
| BMA | 0.23 | | 0.83 | 0.85 | 0.91 |
| BART | 0.18 | 0.17 | | 0.69 | 0.98 |
| XGB | 0.18 | 0.15 | 0.31 | | 0.76 |
| ELN | 0.17 | 0.09 | 0.02 | 0.24 | |

Table 74: 1-step forecast

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.13 | 0.89 | 0.88 | 0.86 |
| BMA | 0.87 | | 0.96 | 0.96 | 0.95 |
| BART | 0.11 | 0.04 | | 0.51 | 0.23 |
| XGB | 0.12 | 0.04 | 0.49 | | 0.20 |
| ELN | 0.14 | 0.05 | 0.77 | 0.80 | |

Table 75: 2-step forecast

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.13 | 0.91 | 0.89 | 0.88 |
| BMA | 0.87 | | 0.94 | 0.93 | 0.93 |
| BART | 0.09 | 0.06 | | 0.27 | 0.16 |
| XGB | 0.11 | 0.07 | 0.73 | | 0.54 |
| ELN | 0.12 | 0.07 | 0.84 | 0.46 | |

Table 76: 3-step forecast

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.06 | 0.91 | 0.96 | 0.93 |
| BMA | 0.94 | | 0.96 | 0.96 | 0.96 |
| BART | 0.09 | 0.04 | | 0.82 | 0.21 |
| XGB | 0.04 | 0.04 | 0.18 | | 0.03 |
| ELN | 0.07 | 0.04 | 0.79 | 0.97 | |

Table 77: 4-step forecast

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.04 | 0.93 | 0.96 | 0.96 |
| BMA | 0.96 | | 0.98 | 0.98 | 0.98 |
| BART | 0.07 | 0.02 | | 0.38 | 0.32 |
| XGB | 0.04 | 0.02 | 0.62 | | 0.42 |
| ELN | 0.04 | 0.02 | 0.68 | 0.58 | |

Table 78: 5-step forecast

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.03 | 0.87 | 0.86 | 0.86 |
| BMA | 0.97 | | 0.99 | 0.98 | 0.98 |
| BART | 0.13 | 0.01 | | 0.10 | 0.04 |
| XGB | 0.14 | 0.02 | 0.90 | | 0.82 |
| ELN | 0.14 | 0.02 | 0.96 | 0.18 | |

Table 79: 6-step forecast

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.10 | 0.84 | 0.82 | 0.81 |
| BMA | 0.90 | | 0.98 | 0.98 | 0.98 |
| BART | 0.16 | 0.02 | | 0.07 | 0.18 |
| XGB | 0.18 | 0.02 | 0.93 | | 0.49 |
| ELN | 0.19 | 0.02 | 0.82 | 0.51 | |

Table 80: 7-step forecast

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.05 | 0.83 | 0.81 | 0.80 |
| BMA | 0.95 | | 0.97 | 0.96 | 0.96 |
| BART | 0.17 | 0.03 | | 0.06 | 0.20 |
| XGB | 0.19 | 0.04 | 0.94 | | 0.48 |
| ELN | 0.20 | 0.04 | 0.80 | 0.52 | |

Table 81: 8-step forecast

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.01 | 0.97 | 0.86 | 0.97 |
| BMA | 0.99 | | 0.99 | 0.99 | 0.99 |
| BART | 0.03 | 0.01 | | 0.00 | 0.02 |
| XGB | 0.14 | 0.01 | 1.00 | | 1.00 |
| ELN | 0.03 | 0.01 | 0.98 | 0.00 | |

Table 82: 9-step forecast

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.01 | 0.84 | 0.80 | 0.84 |
| BMA | 0.99 | | 0.98 | 0.97 | 0.98 |
| BART | 0.16 | 0.02 | | 0.01 | 0.39 |
| XGB | 0.20 | 0.03 | 0.99 | | 0.88 |
| ELN | 0.16 | 0.02 | 0.61 | 0.12 | |

Table 83: 10-step forecast

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.01 | 1.00 | 0.97 | 0.94 |
| BMA | 0.99 | | 0.99 | 0.99 | 0.99 |
| BART | 0.00 | 0.01 | | 0.01 | 0.06 |
| XGB | 0.03 | 0.01 | 0.99 | | 0.00 |
| ELN | 0.06 | 0.01 | 0.94 | 1.00 | |

Table 84: 11-step forecast

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.03 | 0.99 | 0.96 | 0.56 |
| BMA | 0.97 | | 0.97 | 0.97 | 0.96 |
| BART | 0.01 | 0.03 | | 0.02 | 0.18 |
| XGB | 0.04 | 0.03 | 0.98 | | 0.28 |
| ELN | 0.44 | 0.04 | 0.82 | 0.72 | |

Table 85: 12-step forecast

Table 86: Compare for description of table 15. Sector: *NFC-RE*. Source: Deutsche Bundesbank, Bundesbank's credit register, 2008 until 2022, Own calculation

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.12 | 0.22 | 0.77 | 0.93 |
| BMA | 0.88 | | 0.88 | 0.89 | 0.89 |
| BART | 0.78 | 0.12 | | 0.99 | 1.00 |
| XGB | 0.23 | 0.11 | 0.01 | | 0.93 |
| ELN | 0.07 | 0.11 | 0.00 | 0.07 | |

Table 87: 1-step forecast

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.17 | 0.83 | 0.85 | 0.84 |
| BMA | 0.83 | | 0.86 | 0.86 | 0.86 |
| BART | 0.17 | 0.14 | | 0.92 | 0.81 |
| XGB | 0.15 | 0.14 | 0.08 | | 0.55 |
| ELN | 0.16 | 0.14 | 0.19 | 0.45 | |

Table 88: 2-step forecast

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.17 | 0.85 | 0.85 | 0.80 |
| BMA | 0.83 | | 0.83 | 0.83 | 0.83 |
| BART | 0.15 | 0.17 | | 0.49 | 0.29 |
| XGB | 0.15 | 0.17 | 0.51 | | 0.30 |
| ELN | 0.20 | 0.17 | 0.71 | 0.70 | |

Table 89: 3-step forecast

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.18 | 0.90 | 0.94 | 0.87 |
| BMA | 0.82 | | 0.82 | 0.82 | 0.82 |
| BART | 0.10 | 0.18 | | 0.29 | 0.65 |
| XGB | 0.06 | 0.18 | 0.71 | | 0.70 |
| ELN | 0.13 | 0.18 | 0.35 | 0.30 | |

Table 90: 4-step forecast

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.19 | 0.91 | 1.00 | 0.75 |
| BMA | 0.81 | | 0.81 | 0.81 | 0.81 |
| BART | 0.09 | 0.19 | | 0.30 | 0.25 |
| XGB | 0.00 | 0.19 | 0.70 | | 0.46 |
| ELN | 0.25 | 0.19 | 0.75 | 0.54 | |

Table 91: 5-step forecast

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.19 | 0.88 | 0.91 | 0.81 |
| BMA | 0.81 | | 0.81 | 0.81 | 0.81 |
| BART | 0.12 | 0.19 | | 0.19 | 0.18 |
| XGB | 0.09 | 0.19 | 0.81 | | 0.54 |
| ELN | 0.19 | 0.19 | 0.82 | 0.46 | |

Table 92: 6-step forecast

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.20 | 0.98 | 0.93 | 0.89 |
| BMA | 0.80 | | 0.80 | 0.80 | 0.80 |
| BART | 0.02 | 0.20 | | 0.20 | 0.19 |
| XGB | 0.07 | 0.20 | 0.80 | | 0.41 |
| ELN | 0.11 | 0.20 | 0.81 | 0.59 | |

Table 93: 7-step forecast

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.20 | 0.87 | 0.68 | 0.81 |
| BMA | 0.80 | | 0.80 | 0.80 | 0.80 |
| BART | 0.13 | 0.20 | | 0.21 | 0.18 |
| XGB | 0.32 | 0.20 | 0.79 | | 0.39 |
| ELN | 0.19 | 0.20 | 0.82 | 0.61 | |

Table 94: 8-step forecast

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.21 | 0.99 | 0.95 | 0.96 |
| BMA | 0.79 | | 0.79 | 0.79 | 0.79 |
| BART | 0.01 | 0.21 | | 0.27 | 0.22 |
| XGB | 0.05 | 0.21 | 0.73 | | 0.42 |
| ELN | 0.04 | 0.21 | 0.78 | 0.58 | |

Table 95: 9-step forecast

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.22 | 0.99 | 0.94 | 0.96 |
| BMA | 0.78 | | 0.78 | 0.78 | 0.78 |
| BART | 0.01 | 0.22 | | 0.26 | 0.14 |
| XGB | 0.06 | 0.22 | 0.74 | | 0.31 |
| ELN | 0.04 | 0.22 | 0.86 | 0.69 | |

Table 96: 10-step forecast

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.22 | 0.97 | 0.96 | 0.72 |
| BMA | 0.78 | | 0.78 | 0.78 | 0.78 |
| BART | 0.03 | 0.22 | | 0.23 | 0.15 |
| XGB | 0.04 | 0.22 | 0.77 | | 0.19 |
| ELN | 0.28 | 0.22 | 0.85 | 0.81 | |

Table 97: 11-step forecast

| | MBMA | BMA | BART | XGB | ELN |
|---|---|---|---|---|---|
| MBMA | | 0.23 | 0.91 | 0.90 | 0.87 |
| BMA | 0.77 | | 0.77 | 0.77 | 0.77 |
| BART | 0.09 | 0.23 | | 0.10 | 0.06 |
| XGB | 0.10 | 0.23 | 0.90 | | 0.18 |
| ELN | 0.13 | 0.23 | 0.94 | 0.82 | |

Table 98: 12-step forecast

Table 99: Compare for description of table 15. Sector: *NFC-nonRE*. Source: Deutsche Bundesbank, Bundesbank's credit register, 2008 until 2022, Own calculation

# References


Burnham, K. P., and Anderson, D. R. (2002). *Model selection and multimodel inference: A practical information-theoretic approach*. Springer.

Chen, T., and 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., and McCulloch, R. E. (1998). Bayesian cart model search. *Journal of the American Statistical Association*, *93*(443), 935–948.

Chipman, H. A., George, E. I., and McCulloch, R. E. (2010). Bart: Bayesian additive regression trees.

Clyde, M., and George, E. I. (2004). Model uncertainty.

Dorie, V., Perrett, G., Hill, J. L., and Goodrich, B. (2022). Stan and bart for causal inference: Estimating heterogeneous treatment effects using the power of stan and the flexibility of machine learning. *Entropy*, *24*(12), 1782.

Fisher, A., Rudin, C., and Dominici, F. (2019). All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously. *Journal of Machine Learning Research*, *20*(177), 1–81.

Furnival, G. M., and Wilson, R. W. (2000). Regressions by leaps and bounds. *Technometrics*, *42*(1), 69–79.

Gneiting, T. (2011). Making and evaluating point forecasts. *Journal of the American Statistical Association*, *106*(494), 746–762.

Gneiting, T., and Ranjan, R. (2011). Comparing density forecasts using threshold-and quantile-weighted scoring rules. *Journal of Business & Economic Statistics*, *29*(3), 411–422.

Guth, M. (2022). Predicting default probabilities for stress tests: A comparison of models. *arXiv preprint arXiv:2202.03110*.

Harvey, D., Leybourne, S., and Newbold, P. (1997). Testing the equality of prediction mean squared errors. *International Journal of forecasting*, *13*(2), 281–291.

Leamer, E. E. (1978). Regression selection strategies and revealed priors. *Journal of the American Statistical Association*, *73*(363), 580–587.

Madigan, D., and Raftery, A. E. (1994). Model selection and accounting for model uncertainty in graphical models using occam's window. *Journal of the American Statistical Association*, *89*(428), 1535–1546.

McDonald, G. C. (2009). Ridge regression. *Wiley Interdisciplinary Reviews: Computational Statistics*, *1*(1), 93–100.

McDonald, G. C., and Schwing, R. C. (1973). Instabilities of regression estimates relating air pollution to mortality. *Technometrics*, *15*(3), 463–481.

Raftery, A. E. (1995). Bayesian model selection in social research. *Sociological methodology*, 111–163.

Siemsen, T., and Vilsmeier, J. (2017). A stress test framework for the german residential mortgage market: Methodology and application.

Siemsen, T., and Vilsmeier, J. (2018). On a quest for robustness: About model risk, randomness and discretion in credit risk stress tests.

Taggart, R. (2022). Evaluation of point forecasts for extreme events using consistent scoring functions. *Quarterly Journal of the Royal Meteorological Society*, *148*(742), 306–320.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, *58*(1), 267–288.

Tierney, L., and Kadane, J. B. (1986). Accurate approximations for posterior moments and marginal densities. *Journal of the american statistical association*, *81*(393), 82–86.

Varian, H. R. (1975). A bayesian approach to real estate assessment. *Studies in Bayesian econometrics and statistics in honor of Leonard J. Savage*.

Vasicek, O. (2002). The distribution of loan portfolio value. *Risk*, *15*(12), 160–162.

Zou, H., and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, *67*(2), 301–320.

## Declaration

I, Maximilian Hennig, matriculation number 3206100, hereby declare that this Master's thesis, entitled

Prediction of Default Probabilities in the Context of Credit Risk Stress Testing: A Comparison of Methods

is the result of my own independent work. All sources and auxiliary materials used have been properly cited and acknowledged. This thesis has not been submitted to any other examination authority and has not been published elsewhere.

Marburg, 18 June 2025

Maximilian Hennig