Forecasting Recessions in Germany with Feature Selection and Machine Learning

Philip Rademacher

Suggested Citation:
Rademacher, P. (2025). Forecasting Recessions in Germany with Feature Selection and Machine Learning. Journal of Business Cycle Research , 21(2–3), 119–157.
https://doi.org/10.1007/s41549-025-00115-0

Journal of Business Cycle Research (2025) 21:119–157
https://doi.org/10.1007/s41549-025-00115-0

RESEARCH PAPER

Forecasting Recessions in Germany with Feature Selection and Machine Learning

Philip Rademacher[1]

Received: 20 November 2024 / Accepted: 23 October 2025 / Published online: 7 November 2025

Abstract
This study evaluates whether feature selection improves machine learning forecasts of German business cycles. Using a high-dimensional dataset with 73 indicators, primarily from the OECD Main Economic Indicator Database, covering a period from 1973 to 2023, Sequential Floating Forward Selection (SFFS) is applied to build compact, explainable, and performant models. The focus is on regularized regression models (LASSO, Ridge, Elastic Net) and tree-based classification models (Random Forest, Gradient Boosting and AdaBoost). SFFS yields models with up to eleven indicators that outperform a standard term-spread probit model—especially during Quantitative Easing. Regularized regressions provide the most accurate recession signals. Feature selection increased the forecasting power of tree-based models, while marginally reducing the performance of regression models. The findings contribute to the ongoing discussion on the use of machine learning in economic forecasting, especially in the context of limited and imbalanced data.

Keywords Business Cycles · Recession · Forecasting · Machine Learning

JEL Classifcation C52 · C55 · E32 · E37

1 Introduction

After the German economy had just recovered from its pandemic-related downturn, the risk of a new recession has risen dramatically since the war in Ukraine. Recessions represent severe economic downturns that are associated with significant welfare losses, making them an act of strength for the population and policymakers. For

✉ Philip Rademacher
philip.rademacher@hhu.de

[1] Heinrich-Heine University Düsseldorf, Düsseldorf, Germany

this reason, many papers have analyzed indicators and methods that can be used to forecast recessions at an early stage.

Since the seminal study by Estrella and Hardouvelis (1991), in economic research, probit models have traditionally been the tool of choice. Many research contributions confirmed the predictive power of the term spread in other countries (see e.g. Chinn & Kucko, 2015) or improved the models by adding more indicators, such as interest rates, stock prices, bond spreads, price trends or survey-based indicators on consumer or business confidence (e.g., Estrella & Mishkin, 1998, Gilchrist & Zakrajsek 2012, Christiansen et al., 2014, Pönka, 2017, Hasse & Lajaunie, 2020, Nissilä, 2020). Simultaneously, researchers discussed methodological extensions of static forecast models, such as the dynamic model (Nyberg, 2010) that accounts for the autocorrelation structure, and a business-cycle-specific model (Chauvet & Potter, 2005) that allows the variance to vary across cycles.

However, probit models have the disadvantage that they can overfit quickly when using many indicators simultaneously. Therefore, only a few indicators are analyzed in combination, or high dimensional datasets are reduced to common factors and then included in models. Examples of this approach include the works of Christiansen et al. (2014), Chen et al. (2011), and Pönka (2017).

In recent years, there has been a shift towards the application of machine-learning (ML). The higher forecasting power of ML-techniques compared to traditional Probit models is commonly attributed to their ability to capture non-linear relationships and interaction effects between indicators. Furthermore, ML-models are often praised for their higher resilience to overfitting, as they can process many indicators and identify the most important ones.

The literature covers methods such as boosting (Ng, 2012; Berge, 2015; Döpke et al., 2015, 2017; Nevasalmi, 2022; Cadahia Delgado et al., 2022), regularized logistic regression (Choi et al., 2023), and support vector machines (Plakandaras et al., 2017), as well as large-scale comparisons of different methods (Vrontos et al., 2021; Yazdani, 2020; Psimopolous, 2020; Zyatkov & Krivorotko, 2021). Although many contributions refer to the USA, a direct comparison of the results is not possible due to the different datasets and research settings. However, in all studies, one ML-technique outperformed the benchmark. For example, Vrontos et. al. (2021) achieved the best predictions with penalized logit regression, k-nearest neighbours, and Bayesian generalized linear models, while Yazdani (2020) with Random Forest and Zyatkov and Krivorotko (2021) with neural networks. Applications for Germany are the contributions by Döpke et al., (2015, 2017), that applied Logitboost, and Psimopoulus (2020), which implemented several tree-based ensemble methods as well as k-nearest-neighbors and support vector machines.

Although machine learning models can use a large set of indicators, many research papers state that there are a few indicators that have strong forecasting power. In fact, the term spread has long been considered a stand-alone early-warning signal (Chinn & Kucko, 2015). Other studies pointed out the high predictive power of specific indicators, such as stock indices (e.g., Estrella & Mishkin, 1998; Nyberg, 2010; Yazdani, 2020), bond spreads (Ng, 2014) or business and consumer confidence indicators (Christiansen et al., 2014). When using larger datasets and ML methods, fea-

ture importance often confirmed that predictive power is concentrated among a few indicators (e.g., Ng, 2014; Nevasalmi 2021; Vrontos et al., 2021; Choi et al., 2023).

In general, there is no debate as to whether, for the purpose of recession forecasting with ML methods the processing of high-dimensional data is beneficial. Studies currently available in the field of recession forecasting incorporate all indicators from their datasets and do not discuss feature selection, as a dimensionality reduction process that discards redundant, irrelevant, or noisy variables. This gap is particularly notable, given that feature selection can enhance forecasting accuracy, although such improvements are not guaranteed. (Wang et al., 2017). Furthermore, feature selection enhances the interpretability of ML models and reduces the maintenance costs, as the model is dependent on the provision of fewer indicators (Zhao et al., 2019).

This article provides an empirical analysis of how machine learning techniques and feature selection can predict recessions in Germany. Therefore, an extensive dataset comprising 73 indicators mainly from financial markets and the OECD Main Economic Indicators database is compiled. Various models are trained both with and without feature selection. The model-agnostic Sequential Forward Floating Selection (SFFS) algorithm is applied to identify relevant indicators. To examine, the gain in forecasting power that can be achieved through this kind of feature selection, the out-of-sample performance of the models with feature selection is compared to several benchmark models.

By focusing on Germany—where empirical evidence has been less prevalent than the extensive analysis of US datasets—this paper makes a significant empirical contribution to the field of recession forecasting. The three main research questions are: (1.) How do frequently used machine-learning algorithms perform at predicting recessions in Germany? (2.) Can machine-learning forecasts be improved by applying feature selection with SFFS? (3.) Which features are selected by the SFFS algorithm and how do they differ from well-known indicators from the literature?

The paper begins by describing the dataset and defining the recession criteria used in this study (chapter 2). It then explains the classification methods (chapter 3), offers an overview of the empirical strategy (chapter 4), and ends by presenting and discussing the results (chapter 5).

2 Data

2.1 Recessions

The concept of recessions lacks a universally accepted definition. Heileman (2019) emphasizes an easily overlooked fact: business cycles and recessions are statistical constructs, whose temporal classifications vary greatly depending on the underlying definition. This is why numerous definitions exist side by side.

The National Bureau of Economic Research (NBER), a research institute that dates business cycles in the United States, defines a recession as "a significant decline in economic activity spread across the economy, lasting more than a few months, normally visible in real GDP, real income, employment, industrial production, and

wholesale-retail sales." (NBER, 2008). Their definition is widely used in the empirical literature for the USA.

The present article uses the business cycles released by the German Council of Economic Experts (GCEE), whose approach is closely related to the NBER's methodology (GCEE, 2023). In Germany, they identified seven recessions for the period from 1950 to 2023. The last recession started in the last quarter of 2019 and overlapped with the lockdown during the coronavirus pandemic in 2020. Figure 1 provides an overview of the recession phases and also shows their relationship to GDP growth. One advantage of this definition of recession is that they provide the exact start and end month of a recession, e.g. the recession in 1982 lasted from January to November. This enables us to forecast recessions in a monthly setting.

The recession definitions provided by expert boards, such as the GCEE, are not strictly rule-based and therefore contain an element of subjectivity. An alternative concept is the technical recession, which occurs when GDP decreases in two subsequent quarters. Both concepts of recession overlap to some extent, as Fig. 1 shows. It is noticeable that the GCEE recessions start earlier than the decline in GDP, namely immediately after the peak of the growth rates, and end at their trough.

In addition, there are some expansion phases of very low growth or GDP decline that are not labeled as recessions by the GCEE, but as expansion phases. The period 2012–2014 is very striking, as GDP remained nearly constant and declined in two quarters, but according to the GCEE, this period was not a recession phase. (Breuer et al., 2018) Such subjective aspects of the recession definition can cause problems for model training, which will be discussed later. In this article, however, the term "expansion phase" is referred to for all periods that are not explicitly categorized as recessions by the GCEE.

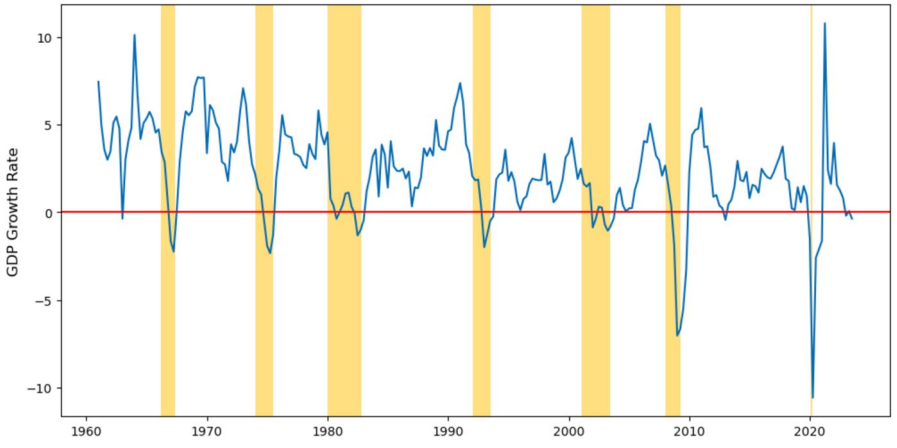


Fig. 1 The shaded periods indicate recessions as classified by the German Council of Economic Experts. The line shows the GDP growth rates compared to the same period in the previous year

2.2 Indicators

To forecast recessions, this paper uses a large set of predictors, predominantly from the OECD Main Economic Indicators Database. The indicators in the analysis should fulfil the following requirements: They should be (1) available for a long period of time, (2) released monthly and (3) not subject to major revisions. (See also: Döpke 2015, p. 43).

The first requirement is applied to ensure that the dataset has as many past recessions as possible. Setting a starting year can be seen as a trade-off between the availability of explanatory indicators and the number of recessions included in the time series. For example, setting 2000 as the starting year would provide access to over 500 indicators but limit the data to three recessions. To ensure that the models can be trained and tested on multiple recession periods, 1973 was selected as the starting year. The resulting time series covers six recessions, and the Consumer Confidence Indicator, a well-known recession indicator, is available for the whole study period. Indicators that are not available for the entire period from 1973 to 2020 are excluded from the analysis.

The restriction on monthly data results from the decision to forecast recessions in a monthly frequency. The monthly setting leads to three times more observations—compared to a quarterly setting—and therefore more training and test material for the models. Some contributions augment their monthly datasets with indicators that are published quarterly. For example, they interpolate quarterly data to a monthly frequency or just forward fill the value from the last quarter. However, the analysis here is restricted to the monthly indicators from the MEI database. The main reason for this decision is that the monthly indicators already cover a wide range of economic topics, and the inclusion of quarterly indicators from the MEI would not increase the diversity of topics.

The third requirement aims to prevent the use of future information in forecasting, also known as look-ahead bias. Ideally, forecasts should rely exclusively on information available at the time of prediction. However, when the dataset includes indicators that are revised or not published in real time, forecasts may inadvertently use information that was unavailable at the forecast's point in time. To address this problem, the initially uploaded data from 2021–2023 was analyzed, to identify indicators frequently published with delays. The average delay for each indicator was calculated and rounded to whole numbers. Indicators with an average delay of three months or more were regarded as not published in real-time and therefore excluded from the sample. Indicators commonly subject to major revisions—such as GDP growth rates, industrial production, imports and exports —were additionally removed from the sample.

The resulting cleaned data set should be understood as closer to real-time data, but does not represent the original data that was available when the forecasts should be made. Some indicators may still have been available late during some periods and minor data revisions may also have taken place. In addition, some indicators of the MEI are only available seasonally adjusted.[1] This can also lead to data leak-

[1] For an overview refer to “List of Indicators” in the Appendix.

age, potentially introducing a small look-ahead bias. Nevertheless, this limitation is accepted, as the remaining bias is assumed to be minimal. Additionally, since the primary focus of this paper is the comparison of different models (with and without feature selection), it is assumed that all models are equally affected by any bias, leaving the comparison itself unaffected (c.f. Kant et al., 2025, p. 12).

Some important financial indicators, that are out of scope of the MEI database, were added to the study sample due to their frequent use in literature. These include two stock indices, namely the DAX and the SP 500, oil prices, the US term spread, the transatlantic term spread, i.e., the difference between the German and US term spread, and some US bond spreads. Although the latter primarily reflects US economic trends, some contributions have observed a predictive power for recessions in other countries. (e.g., Nissilä, 2020). The US term spread has frequently been used to successfully forecast recessions in other countries, and the transatlantic term spread has proven to be a reliable indicator to predict recessions in Germany (Nyberg, 2010). As the DAX was first launched in 1988, retrospectively calculated values were used for the previous period. (Stehle et al., 1996, p. 24).

In general, indicators are transformed to differences or percentage changes relative to the previous year. Exceptions are indicators whose values have a constant scale over time, such as interest rates or survey-based indicators with a given scale. These are also included as levels. Indicators that were indexed to a certain year are only included as percentage changes in the study sample.

3 Classifiers

Predicting recessions is a classification task, as the recession indicator y_t is a binary variable with the values:

$$y_t = \begin{cases} 1, & \text{if recession in period } t \\ 0, & \text{if expansion in period } t \end{cases} \tag{1}$$

All classification models estimate the probability of a recession occurring within one year in the future, as a function of some indicators X_{t-h} at the time of the forecasts. The models are calibrated with a threshold of 50%, i.e. probabilities higher than this value indicate a future recession.

$$\Pr\left(y_t = 1 | X_{t-h}\right) \tag{2}$$

if $\Pr(y_t > 0.5 | X_{t-h})$ then $\widehat{y}_t = 1$

Two groups of classifiers will be used to forecast recessions, namely generalized linear regression models, and tree-based ensemble models. Generalized linear models have the advantage of being easy to interpret due to their coefficients, but they are restricted by their functional form. Interaction effects and non-linearities must be specified ex ante. Tree-based methods, on the other hand, can model non-linear relationships and interaction effects between indicators easily due to their non-parametric structure.

However, all classifiers selected for this paper provide measures to interpret the feature relevance as part of their standard implementation. While the regression models have coefficients, in the tree-based models, feature importance can be calculated based on their contribution to improving the impurity. Classifiers, such as K-nearest neighbours, SVM or Neural Networks, are more difficult to interpret as this requires model-agnostic tools to analyse the effect of features.

The latter models are out of scope here for two reasons: First the focus is on simple, compact and interpretable models. Second, this comprehensible structure of feature importance is technically helpful to calculate a further group of benchmark models. In these models, feature selection is based on the most significant features of the full models, whereby Gini importance or coefficients are used.

3.1 Regression Models

Logistic regression and probit regression are used as benchmark models in this paper. Both models are parametric, i.e. they parameterize the relationship between the indicators and the recession indicator in a linear way with coefficients ($\widehat{y} = X\prime\beta$.) The main difference to the linear model is that both binary regression models use a sigmoid function, ensuring the predictions are probabilities bounded between 0 and 1.

$$\Lambda\left(X'\beta\right) = \frac{e^{X'\beta}}{1 + e^{X'\beta}} \tag{3}$$

$$\Phi\left(X'\beta\right) = \int_{-\infty}^{X'\beta} \phi\left(z\right) dz \tag{4}$$

The logistic regression uses the cumulative distribution function of the logistic distribution (Eq. 3) as the sigmoid function, while the probit model uses the distribution function of the standard normal distribution (Eq. 4). As a result, both regression models lack a closed form and therefore the estimation is done iteratively with maximum likelihood.

$$\mathcal{L}_N\left(\beta\right) = \sum_{t=1}^{T} \left\{y_t \ln F\left(x_{t-h}\prime\beta\right) + \left(1 - \mathrm{y}_t\right) \ln(1 - F\left(x_{t-h}\prime\beta\right))\right\} \tag{5}$$

Optimal coefficient estimates are obtained by maximizing the log-likelihood function ($\mathcal{L}_N$) defined in Eq. 5. Note that, the sigmoid function, which varies between logit and probit regression, is labelled as $F(x_{t-h}\prime\beta)$.

In this study, probit regression is exclusively used to replicate a benchmark-model commonly found in prior literature, that uses the term spread as a single predictor. Logistic regression is used as a benchmark model with all features. Both models are not considered as ML and therefore not challenged with the SFFS.

As recessions occur less frequently in the data than expansion phases, the logistic regression is implemented with class weights.[2] Therefore, a weight w_t is assigned to each period in the likelihood function, whereby periods 12 months prior to a recession are given a higher weight (Eq. 6.). For example, if there are 40 expansion and 20 recession phases in a data set, periods prior to a recession are given the weight $w_R = 1.5$ (Eq. 7).

$$\mathcal{L}_N(\beta) = \sum_{t=1}^{T} w_{t-h} * \{y_t \ln F(x_{t-h}\prime\beta) + (1 - \mathrm{y}_t) \ln(1 - F(x_{t-h}\prime\beta))\} \tag{6}$$

$$w_{t-h} \in \{w_R, w_E\}$$

$$w_R = \frac{T}{2 * T_R} \; w_E = \frac{T}{2 * T_E} \tag{7}$$

$$T = T_R + T_E$$

As the role of feature selection in ML is analysed, three regularized variants of the logistic regression are employed, namely LASSO, Ridge and Elastic Net. All of them augment the objective function ($\mathcal{L}_N$) with a penalty term, that controls the size of the coefficients. This penalizing aims to control model complexity and mitigate overfitting. Typically, this also improves the out-of-sample performance. Concretely, LASSO (Tibshirani, 1996) minimizes the sum of the absolute coefficients, while Ridge (Hoerl & Kennard, 1970) minimizes the sum of the squared coefficients instead. Elastic Net (Zou & Hastie, 2005) combines both penalties. As a result, in those models compared to a standard logistic regression, many coefficients are small or close to zero. Equation 8 shows a setting with penalties on the coefficients; depending on the values for λ_L and λ_L a logistic regression, LASSO, Ridge or Elastic Net model is obtained. The optimal regularization strength, i.e. the specific values for λ_L and λ_R, are defined in the hyperparameter tuning.[3]

$$\max_{\beta} \mathcal{L}_N - \lambda_L \sum_{i=1}^{p} |\beta_i| - \lambda_R \sum_{i=1}^{p} (\beta_i)^2 \tag{8}$$

$\lambda_L = 0, \lambda_R = 0$ Logistic Regression.
$\lambda_L \neq 0, \lambda_R = 0$ LASSO.
$\lambda_L = 0, \lambda_R \neq 0$ Ridge.
$\lambda_L \neq 0, \lambda_R \neq 0$ Elastic Net.

[2] As common in the literature, the probit benchmark model is not weighted.

[3] For an overview of the tuned hyperparameters and the search spaces, refer to "Hyperparameters and Search Spaces" in the appendix.

3.2 Tree Based Models

The foundation of the following tree-based models is the simple classification tree, which is also used as a benchmark model. The classification tree recursively partitions the dataset into a collection of disjoint regions (R), assigning a class label to each region. At each split, a straightforward decision rule is applied. For instance, the tree might first decide that "Term Spread$\leq$0" is the best rule, creating two regions: one where recessions predominate (Term Spread$\leq$0) and one where expansions predominate (Term Spread$>$0). Each of those regions can then be split again, ultimately yielding four regions in the next step.

At every node, the algorithm selects the feature and split point that most improve classification accuracy, based on the Gini impurity GI (Eq. 9). The Gini impurity ranges from 0, when a region is perfectly pure and consists solely of one class, to 0.5, when classes within a region are equally mixed. As an example, suppose a region consists of 85% expansion periods. Its Gini impurity would be calculated as $2(0.85 \times 0.15) = 0.25$, reflecting a moderate level of class mixing.

$$GI\left(R\right) = 2(p\left(1-p\right)) \tag{9}$$

The objective of each split (with feature j and split point s) is to decrease Gini impurity. Therefore, the impurity of the new regions is compared against the impurity of the parent region (or, for the very first split, the entire dataset). Equation 10 describes, how the decrease for each candidate region is calculated. For $I(R)$ the Gini Impurity $GI(R)$, Eq. 9 is used. Note that region size matters: isolating small, already-pure subsets can yield smaller impurity gains than splitting larger, more mixed regions. The algorithm repeats this process until a defined stop-criterion is met (e.g. max depth or max leaf size is reached) or no further decrease in impurity is possible.

$$I\left(j,s\right) = I\left(R\right) - \frac{T_{left}}{T} I\left(R_{left}\right) - \frac{T_{right}}{T} I\left(R_{right}\right) \tag{10}$$

Three hyperparameters are tuned to mitigate overfitting in the classification tree model. These include: (1) the maximum tree depth, which determines the number of successive splits; (2) the minimum number of observations required in a terminal region; and (3) the minimum impurity decrease. The latter is by default zero, meaning any improvement justifies a split. During hyperparameter tuning, this value is increased to enforce stricter splitting criteria, effectively implementing an early-stopping mechanism when the marginal improvement in impurity falls below a predefined threshold.

Consistent with the approach used in logistic regression, weights are implemented for recession and expansion phases (calculated as presented in Eq. 7). The weights have a direct effect on the Gini impurity: If there have been 17 expansions (=85%) and 3 recessions in a region, the unweighted Gini impurity was 0.25. If, however, recession periods are assigned double weight ($w_R = 2, w_E = 1$), the 3 recessions are effectively counted as 6. This adjustment alters the distribution to 17 expansions out of 23 total weighted observations, reducing the expansion share to approximately

73%. Consequently, the Gini impurity increases to 0.394, indicating higher impurity. Therefore, the weights directly affect the optimal splits and thus how the whole classification tree is built.

3.2.1 Random Forest

Random Forest (Breimann, 2001) consists of numerous randomized classification trees (S). The randomness arises from two sources: each tree is trained on a different bootstrap sample of the training data (i.e., random sampling with replacement), and at each split within a tree, a random subset of predictors is considered rather than the full set. This dual randomization ensures that the individual trees are largely independent of each other.
Random Forest is an ensemble method that combines single classification trees with majority voting: Each classification tree gives a prediction and if the majority (e.g. 60%) of the trees predict a recession, the final recession probability would be 60%. This aggregation of multiple weak learners reduces variance and enhances generalization performance, making Random Forest more robust to overfitting compared to a single classification tree (Hastie et al., 2009, pp. 587–588).

Random Forest	-
For $s = 1,2,\ldots S$:	
a) Sample observations with replacement from the training data	
b) Train a classification tree, where each split considers a random subset of all available features	
Classify new observations by taking the majority class vote of the S trees	

In hyperparameter tuning, the number of trees (S), their maximum depth and the number of features and samples are tuned. The weighting in the random forest is applied at the level of individual classification trees, as described in the previous section. The weighting is based on the subsamples, i.e. first the bootstrap samples are generated and then the weights are calculated. The weights therefore differ from tree to tree.

3.2.2 AdaBoost

Adaptive Boosting (AdaBoost), introduced by Freund and Schapire (1997), is another tree-based method that, in contrast to Random Forest, combines the weak learners in a sequential way. Consequently, the weak learners are not independent of each other. As there are some variants of AdaBoost, it is important to note, that this paper refers to the AdaBoost.M1 algorithm known as "Discrete AdaBoost".

The algorithm aims to improve classification performance by reweighting observations after each iteration. Initially, all observations are assigned equal weights (see pseudo-code box, step 1). A weak learner, technically a classification tree with two terminal regions (labelled as $G_m(x)$), is fitted to the classification problem (step 2a.). The predictions of the weak learner are compared with the actual values and an error rate err_m is calculated. Therefore, the sum of the weights of the misclassified observations is divided by the sum of the weights of all observations (step 2b). Then the amount of say (α_m) is calculated for the weak learner, which determines how strongly this learner influences the overall prediction. α_m is inverse to the error and scaled with

the learning rate λ (step 2c). After each iteration, the weights of misclassified observations are increased, so that subsequent weak learners focus more on the difficult cases. If an observation has been correctly classified, then $I\left(y_t \neq G_m\left(x_{t-h}\right)\right) = 0$ and the weight does not change, since $e^{\alpha_m I\left(y_t \neq G_m\left(x_{t-h}\right)\right)} = 1$. If an observation was misclassified, α_m determines how strong the weight increases in the next iteration. If a weak learner classifies well overall, a misclassification is weighted stronger than a misclassification in the case of a poor weak learner.
The final prediction is a weighted majority vote of the individual classifiers, where each model's vote is proportional to its amount of say (step 3). If $G\left(x_{t-h}\right) > 0.5$ than a recession is predicted.

Ada Boost (Hastie et al., 2009, p. 339) -

1. Initialize the observation weights $w_{t-h} = \frac{1}{T}, t = 1, 2, \ldots, T$
2. For $m = 1$ to M:
a) Fit a classifier $G_m\left(x\right)$ to the training data using weights w_t
b) Compute

$$err_m = \frac{\sum_{t=1}^{T} w_{t-h} I\left(y_t \neq G_m\left(x_{t-h}\right)\right)}{\sum_{i=1}^{T} w_{t-h}}$$

c) Compute $\alpha_m = \lambda * \log\left(\frac{1-err_m}{err_m}\right)$
d) Set $w_i \leftarrow w_{t-h} e^{\alpha_m I\left(y_t \neq G_m\left(x_t\right)\right)}$
3. Output $G\left(x\right) = \sum_{m=1}^{M} \alpha_m G_m\left(x\right)$

Like the previous classifiers, weights are introduced to address the imbalanced nature of the data. In AdaBoost sample weights are implemented in the first step: Recessions receive w_R and expansions w_E instead of equal weights. However, in subsequent iterations, these weights are updated as described above. Two hyperparameters are tuned: the number of weak learners M and the learning rate λ.

3.2.3 Gradient Boosting

Gradient Boosting (Friedman, 2001) generalizes the boosting idea by fitting each weak learner to the negative gradient of a differentiable loss function, rather than by reweighting misclassified samples as in the discrete version of AdaBoost. The pseudo-code box provides a step-by-step description of the algorithm. In this paper, the logistic loss function is adopted, as defined below in dependence of the predicted probability p and the odds γ.

The algorithm initializes by assigning an identical starting prediction to every observation. The starting prediction ($p_0\left(x_{t-h}\right)$) is chosen to minimize the loss function across the entire dataset, which is given when each observation's recession probability is set to share of recessions.

In the second step, the M weak learners are sequentially trained. Therefore, the negative gradient, i.e. the difference between the actual outcome (y_t) and the recession probability of the previous iteration ($p_{m-1}(x_{t-h})$), is calculated (step 2a). In the first iteration ($m = 1$), the probability of the previous iteration is the start prediction $p_0\left(x_{t-h}\right)$. Assume that the initial start prediction is 30%, then in the first iteration, a recession receives $1 - 0.3 = 0.7$ and an expansion -0.3 as pseudo-residuals.
A regression tree is fitted on these pseudo-residuals (step 2b). The specific predictions of the regression trees are not of interest, but it is important in which leaf region

an observation ends up. Each observation within a leaf node J receives a new odds ratio, which depends on the sum of the probabilities from the previous iteration and the pseudo-residuals (step 2c). The boosting prediction $f_m(x)$, which is expressed in odds, is updated by adding the odds from step 2c multiplied with the learning rate to the previous boosting prediction $f_{m-1}(x)$. The updated probabilities can be calculated based on the updated odd prediction. In the next iteration, these probabilities are used to fit the subsequent weak learner. This continues until the number of learners M is reached. The final prediction (step 3) consists of the start prediction to which all predictions of the weak learners M are added up.

Gradient Boosting (Hastie et al., 2009, p. 361) -

The differentiable loss function is logistic loss:

$L(y_t, p) = -(y_t * log(p) + (1 - y_t) * \log(1 - p))$

$\iff L(y_t, \gamma) = -y_t * \log(\gamma) + \log(1 + e^{\log(\gamma)})$

1. Initialize model with a constant value $f_0(x_{t-h}) = \underset{\gamma}{\operatorname{argmin}} \sum_{t=1}^{T} L(y_t, \gamma)$

$\Leftrightarrow f_0(x_{t-h}) = \log\left(\frac{T_R}{T_E}\right)$

$p_0(x_{t-h}) = \frac{T_R}{T_R + T_E}$

2. For $m = 1$ to M:

a) For $i = 1, .., T$ compute pseudo-residuals:

$r_{im} = -\left[\frac{\delta L(y_t, f(x_{t-h}))}{\delta f(x_{t-h})}\right]_{f=f_{m-1}}$

$\Leftrightarrow r_{im} = y_t - p_{m-1}(x_{t-h})$

b) Fit a regression tree to the targets r_{im} giving terminal regions $R_{jm}, j = 1, 2, \ldots, j_m$

c) For $j = 1, 2, \ldots, J_m$ compute:

$\gamma_{jm} = \underset{\gamma}{\operatorname{argmin}} \sum_{x_i \in R_{jm}} L(y_t, f_{m-1}(x_{t-h}) + \gamma)$

$\Leftrightarrow \gamma_{jm} = \frac{\sum_{x_{t-h} \in R_{jm}} r_{im}}{\sum_{x_{t-h} \in R_{jm}} p_{m-1}(x_{t-h})(1 - p_{m-1}(x_{t-h}))}$

d) Update $f_m(x_{t-h}) = f_{m-1}(x_{t-h}) + \lambda \sum_{j=1}^{J_m} \gamma_{jm} I(x_{t-h} \in R_{jm})$

$p_m(x_{t-h}) = \frac{e^{\log(f_m(x_{t-h}))}}{1 + e^{\log(f_m(x_{t-h}))}}$

3. Output $\widehat{f}(x_{t-h}) = f_M(x_{t-h})$

$\widehat{p}_m(x_{t-h}) = \frac{e^{\log(\widehat{f}(x_{t-h}))}}{1 + e^{\log(\widehat{f}(x_{t-h}))}}$

The following hyperparameters are tuned in Gradient Boosting: The number of weak learners (M), the learning rate (λ), as well as the maximum depth and the minimum impurity decrease of the trees. Weights are implemented on the tree-level analogue to classification trees or Random Forests. The main difference here, is that Gradient Boosting relies on regression trees instead of classification trees. Hence, the optimization criterion is different: Instead of Gini Impurity, the Mean Squared Error (Eq. 11) is inserted for $I(R)$ into Eq. 10. Apart from that, the weighting works in the same way as for the classification trees.

$$MSE(R) = \frac{1}{N} \sum_{i=1}^{N} (y_i - \overline{y})^2 \tag{11}$$

4 Empirical Design

In the main setting, the time series is divided into a training and a hold-out sample. The training sample spans from 1973 to December 2005 and includes four recession phases, while the hold-out sample begins in January 2006 and includes two recession phases. The hold-out sample is only used to evaluate the model's predictive power. This means that 70% of the data is used for training and 30% for testing, which is very common.

In a robustness check, the so-called dynamic setting, an expanding window approach is employed, where the models are retrained after the recessions of 1992, 2001 and 2008. In the first split, models are trained on data preceding 1998 and tested on the subsequent business cycle until 2005. In the second, the training data is expanded until 2005, and the performance is evaluated on the years during the financial crisis. In the last split, the model is trained on data up to the recession in 2008 and tested until 2020.

Both settings rely for the performance evaluation exclusively on out-of-sample data that remain entirely withheld from all stages of model training, including feature selection and hyperparameter tuning. Emphasizing this design is essential, since alternative "hybrid" approaches in the literature allow information from test data to influence model tuning, thereby introducing the risk of inadvertent data leakage (for a discussion, see Puglia and Tucker (2020, 2021)).

4.1 Model Training

Model training takes place in three steps. In the first step, relevant features are selected, then conditional on the selected features the models hyperparameters are tuned. In the final step, the models are trained with the selected features and hyperparameters on the entire training data.

The selection of features and hyperparameters is performance-based, which means that models are first trained with the specific set of features or hyperparameters and then evaluated on a new period. As this performance-based selection should take place entirely within the training data set, the training data is divided again into a train and validation period.

Therefore an (inner) expanding window is applied: In the first split, the training fold covers the period from 1974 to 1983, and the validation fold covers the period until 1992. In the second split, the training fold is extended by the validation fold from the first pass, and the validation fold spans the period between 1992 and 2001).[4]

This way, each combination of features from the sequential floating forward selection and the hyperparameters are validated. Finally, the performance metrics of the different validation folds are averaged, and the hyperparameter or feature set with the highest mean performance is selected. The expanding window approach ensures that the chronology of the observations is always retained, which requires features to have a general predictive power, i.e. they need to perform well in more than one recession. A feature should be trainable on one recession period (for example, 1980)

[4] For an illustration refer to "Expanding Window Scheme" in the appendix.

and still accurately forecast a later recession (for example, 1992). Using standard cross-validation with random shuffling at this stage risks leaking recession months from the validation into the training sample. For example, predicting February 1990 might look accurate simply because January or March 1990 data were already used during training.

During hyperparameter tuning and feature selection, the model's performance is measured using the negative Brier score (Eq. 12). The Brier score is the mean squared deviation between the model prediction $\widehat{y}_t$ and the actual state y_t for all observations in the validation fold T. If the predictions are always close to the actual state, the model performance is high, and the Brier score is small. The negative Brier score is therefore maximized.

$$BS = \frac{1}{T}\sum_{t=1}^{T}\left(y_t - \widehat{y}_t\right)^2 \tag{12}$$

The Brier score is not the first choice for imbalanced data, more frequently used in such cases are F1 score or the balanced accuracy. However, the Brier score has the advantage, that performance can be measured even if only one class is present in the validation fold (i.e. only recensions or non-recensions). Due to the few recession phases, it is not feasible to create only validation folds with both classes.

4.2 Sequential Forward Floating Selection

In this study, the embedded ability of the ML models to focus on relevant predictors is augmented by a wrapper method called Sequential Forward Floating Selection (SFFS) (Pudil et al., 1994). This method is particularly useful for classifiers that lack regularization and therefore cannot perform embedded feature selection (Raschka & Mirjalili, 2017).

Let $Y = \{y_j | j = 1,2,\ldots,D\}$ denote a dataset consisting of D features. The objective is to construct a subset X_k of k high-performing features using the SFFS algorithm. Prior to execution, the feature set is initialized as empty, i.e., $X_0 = \varnothing$. In the first iteration, for each feature y_j a classification model is trained, and its performance is evaluated using $J(y_j)$.[5] The feature yielding the highest performance, denoted as x_1, is then added to the feature set, resulting in $X_1 = X_0 + x_1$.

In the subsequent iteration, models are constructed using the current feature set X_1 combined with each of the remaining features $Y - X_1$. The feature that, when added toX_1, maximizes the evaluation function $J(X_1 + x_2)$ is selected, yielding an updated set$X_2 = X_1 + x_2$. This sequential inclusion of features continues in further iterations, until the desired number of features k is selected. It is important to note that there is no guarantee that the performance of X_k will exceed that ofX_{k-1}. With increasing iterations, including additional features may lead to a decline in performance.

[5] The evaluation function is the negative Brier score, described in the previous section.

The description up to this point describes the Sequential Forward Selection algorithm. In the floating variant of the algorithm, after each forward inclusion, it is assessed whether the removal of any feature from the current set X_k improves the evaluation function relative to the previous iteration, i.e., $J\left(X_k - x_r\right) > J\left(X_{k-1}\right)$. To avoid endless loops, where a feature is included and directly removed again, only features other than the last included feature can be removed, i.e. $x_r \neq x_k$. After a feature is removed from the set, the exclusion step is repeated until no more features can be removed. The algorithm then proceeds with the inclusion of the next feature.

In this application, the SFFS algorithm is executed for twenty iterations. Afterwards the iteration is identified, where the performance reached its maximum. The corresponding feature set is used to train the final models.

SFFS has two main advantages: First, it selects features based on their performance within the given classifier. Therefore, the selected features can vary between the classifiers. Non-parametric classifiers can rely on different features than parametric ones. The method is therefore superior to simple filter methods that select features based on univariate measures because it takes into account how a feature performs in the specific classification procedure.

The second advantage is, that the features are selected based on their performance in the validation fold. This is in contrast with the embedded feature selection methods of e.g. Random Forest or LASSO, which are performed in-sample. Even if such classifiers are to some degree able to distinguish important from unimportant features, they can suffer from overfitting. During the selection process in SFFS, an indicator has to prove its forecasting power on the validation folds, which reduces the risk of overfitting. (Guyon & Elisseeff, 2003).

On the other side, there are two weaknesses: First, the procedure is computationally complex and is therefore often referred to as a "brute-force" or "greedy" method. This is mainly since many models are trained and evaluated in each iteration. A second limitation stems from the nested nature of feature sets, i.e. the algorithm develops a strong path dependency and builds local-optimal models Because each feature is chosen based on those already selected, the procedure may miss a superior combination that would emerge only if a different feature had been picked first. However, it should be noted that the floating variant already reduces this path dependency by allowing features to be reconsidered and swapped out. Furthermore, the empirical research showed that such path-dependent algorithms can still yield substantial gains in predictive power. (For example: Jain & Zongker, 1997; Zhao et al., 2019).

4.3 Hyperparameter Tuning

Modern machine learning algorithms often require precise tuning of hyperparameters to achieve optimal performance. For each classifier, the key hyperparameters are identified and associated with an appropriate search range. For instance, the number of trees in Random Forest is optimized by trying models with between 50 and 300 trees.[6]

[6] For an overview of the final hyperparameter values refer to “Tuned Hyperparameters” in the appendix.

Hyperparameter tuning is performed using the Optuna framework (Akiba et al., 2019), which efficiently finds suitable hyperparameter values by evaluating different combinations of hyperparameters in multiple trials. Initially, Optuna samples hyperparameter values from the predefined search space at random, but as more trials are conducted, it leverages the results of prior trials to search more systematically.

Optuna offers two key characteristics that make the search more efficient. First, its median pruner stops trials with inferior performance—compared to previous trials – earlier and thereby saves computational resources. Second, the framework's built-in Tree-structured Parzen Estimator (TPE) enhances optimization by reframing hyperparameter tuning as a density estimation problem (Bergstra et al., 2011). TPE splits past trial results into "good" and "bad" groups based on a performance threshold, estimates the likelihood of each hyperparameter set using nonparametric models, and then prioritizes new hyperparameter sets by selecting those that maximize the ratio of promising to less promising outcomes.

4.4 Out-Of-Sample Evaluation

While the Negative Brier Score is used for feature selection and hyperparameter tuning, additional metrics are employed for out-of-sample evaluation. Here, the focus is on assessing the accuracy of recession forecasts rather than solely the underlying probability estimates. Each model is calibrated using a 50% threshold—meaning that a recession signal is generated only when the predicted probability exceeds 50%. Consequently, most performance metrics are derived from the following confusion matrix components:

- True Positives (TP): Number of correctly predicted recessions
- False Positives (FP): Number of incorrectly predicted recessions
- True Negatives (TN): Number of correctly predicted expansion periods
- False negatives (FN): number of incorrectly predicted expansion periods

Recall (Eq. 13) is the true positive rate and shows the percentage of recessions that the model was able to predict correctly. Conversely, a 70% recall indicates that the model incorrectly classified 30% of the recession months as expansions.

$$REC = \frac{TP}{TP + FN} \tag{13}$$

Precision (Eq. 14) is used to measure the reliability of a recession forecast. It shows the proportion of true recessions to the total number of predicted recessions.

$$PRE = \frac{TP}{TP + FP} \tag{14}$$

The F1 score (Eq. 15) is a composite measure that combines precision and recall using the harmonic mean. This is helpful as there is a trade-off between recall and

precision. When a model predicts all recessions, it often results in a loss of precision, as it needs to label more periods as recessions to achieve a high recall.

$$F1 = 2\frac{PRE * REC}{PRE + REC} \tag{15}$$

To assess model performance in the presence of class imbalances, the balanced accuracy score is also calculated (Eq. 16). Balanced accuracy is defined as the mean share of correctly classified recessions (identical to recall) and correctly classified expansions (also known as specificity). Unlike standard accuracy—which can be misleading in datasets with uneven class distributions—balanced accuracy provides a more meaningful evaluation when one class dominates the data. For instance, in a dataset where 90% of the months are expansion, a model that always predicts expansion would yield an overall accuracy of 90% but only a balanced accuracy of 50%. Scores below 50% indicate performance worse than random guessing, while a score of 0% would imply that every recession is misclassified as an expansion and vice versa.

$$BAC = \frac{1}{2} * \left(\frac{TP}{TP + FN} + \frac{TN}{TN + FP}\right) \tag{16}$$

The AUC (Area Under the ROC Curve) measures the classifier's ability to rank recessions above expansion events. The ROC curve shows the false positive rate (on the x-axis) and true positive rate (also known as recall, on the y-axis) for each possible threshold. The closer the curve is to the left of the bisector, the better the calculated model. The area under the curve can be calculated from this curve as a measure of the model's quality. The larger this area, the better the model. An integral calculation using the chord trapezoid formula for integration yields the following result:

$$AUC = \sum_{i=1}^{n-1} \left(FPR_{i+1} - FPR_i\right) \frac{\left(REC_{i+1} + REC_i\right)}{2} \tag{17}$$

AUC ranges from 0 to 1, whereby values close to 1 describes models with relevant explanatory power model and values of 0.5 random guessing. In practice, a high AUC score can be interpreted as a perfect alignment of recession probabilities and actual recessions: The predicted recession probabilities increase before recessions and fall afterwards.

However, the AUC does not assess whether those probabilities exceed the cut-off value of 50%. As a result, a model can achieve a high AUC but poor other metrics (like precision, recall, f1). This implies that the model would be better if we changed the cut-off value. The difficulty is that the models are trained with a cut-off-value of 50% and when forecasting future recessions, we assume that this cut-off value is optimal and reliable. Changing the cut-off value to improve predictions on the hold-out sample would not be practical, as we would not be able to determine the ideal cut-off value ex ante (look-ahead bias). A good model therefore, must combine both, a high AUC and high other metrics.

An additional evaluation metric, the average prediction delay, quantifies the temporal lag between the intended forecast horizon and the actual issuance of a recession signal (Eq. 18). For a one-year horizon, the delay is defined as the difference, in months, between the time point of the desired forecast (s_R) and the model's first recession signal ($\widehat{s_R}$). Thus, a delay of 0 months indicates a recession signal exactly 12 months before the recession onset, and a delay of 2 months indicates a forecast issued only 10 months before the recession onset. If there are two recessions in the evaluation period, the mean delay is calculated by dividing the sum of all delays by the number of actual recessions.

$$DL = \frac{1}{R}\sum_{r=1}^{R} s_R - \widehat{s_R} \tag{18}$$

4.5 SHAP

SHAP values are calculated to analyze the indicators' impact on recession forecasts over time and to test whether the model performance is affected by structural breaks. Specifically, the aim is to evaluate whether feature importance has changed significantly between training and test samples.[7]

The theory of SHapley Additive exPlanations (SHAP) originates from cooperative game theory and was used to distribute the outcome of a game fairly among a coalition of players (Shapley, 1953). The Shapley value for a player i (Eq. 19) is the sum of the weighted marginal contributions of player i to all possible coalitions S of other players which the player can join. The marginal contribution is the outcome of the coalition including player i $F_{\mathrm{S}\cup\{\mathrm{i}\}}\left(\mathrm{X}_{\mathrm{S}\cup\{\mathrm{i}\}}\right)$ minus the outcome without player i $\mathrm{F_S}\left(\mathrm{X_S}\right)$. The individual marginal contributions are weighted by a fraction. $|\mathrm{F}|!$ represents the number of coalitions that can be formed with all players. $|\mathrm{S}|!$ is the number of players in the coalition and $(|\mathrm{F}| - |\mathrm{S}| - 1)!$ the number of possibilities in which further players can join the coalition after play i has already joined.

$$\phi_i = \sum_{\mathrm{S}\subseteq\mathrm{F}\{\mathrm{i}\}} \frac{|\mathrm{S}|!\,(|\mathrm{F}| - |\mathrm{S}| - 1)!}{|\mathrm{F}|!}\left[\mathrm{F}_{\mathrm{S}\cup\{\mathrm{i}\}}\left(\mathrm{X}_{\mathrm{S}\cup\{\mathrm{i}\}}\right) - \mathrm{F_S}\left(\mathrm{X_S}\right)\right] \tag{19}$$

This theoretical framework was transferred to machine learning as a model-agnostic explainer and is available as a Python package (Lundberg & Lee, 2017). This package provides different approximations, so-called explainers, to calculate Shapley values.

[7] There is a strand of literature that addresses structural breaks from a traditional time-series perspective (e.g., Castle et al., 2015; Hendry, 2006). In contrast, when applying machine learning directly, there is no consensual method that handles structural breaks. This paper follows a common empirical setting used in various ML studies and does not explicitly model structural breaks, addressing them only indirectly through the models validation strategy (e.g., Jardet & Meunier, 2022; Kant et al., 2025; Vrontos et al., 2021). To compensate for this limitation, the Shapley-value analysis is applied here as a diagnostic tool to reveal changes in variable importance over time and to help interpret possible breaks within the adopted ML framework.

As the SFFS models remain relatively small, the exact explainer, which computes Shapley values without any approximation, is employed.

Basically, SHAP values assume that the features are independent of each other. However, some features are clearly interrelated and cannot change their characteristics independently of each other. For example, the term spread can only be negative if short-term interest rates are higher than long-term interest rates. An unemployment rate of 8% cannot occur together with a seasonal unemployment rate of 2% in a given month. Such correlations between features are taken into account in the calculation of SHAP values by adding so-called Owen values (Owen, 1977). The Owen value was originally intended in game theory to consider the a priori formation of unions or coalitions of players who prefer to play together. This concept was later transferred to the domain of ML to respect correlations and similar features. (For an overview, see Li et al., 2024).

5 Results

5.1 Feature Selection

In SFFS, the classification models reach their optimum very quickly. After that, their performance either does not improve anymore or decreases. The performance of Random Forest in the validation samples already starts to decrease after the inclusion of the fifth indicator. Gradient Boosting and Ridge have their peak at 7 indicators. LASSO, Elastic Net and AdaBoost processed a larger number of indicators (namely 10, 10, and 15) before their performance reached a maximum. Basically, the results indicate that there are a few strong indicators that drive up performance at the beginning, but then there are no more that can further improve the predictive power in later iterations.

Figure 2 illustrates how the model's performance improves or deteriorates with the inclusion of more indicators. The points show how many indicators were selected for the given classifier. As a (negative) Brier score closer to zero indicates better model performance, Elastic Net slightly outperforms LASSO followed by Ridge. The tree-based methods, namely Random Forest, Gradient Boosting and AdaBoost have an inferior performance in the validation samples.

Table 1 presents the indicators selected by each model. The regression-based methods—LASSO, Ridge, and Elastic Net—identify energy prices as the primary predictor (denoted as “L”) in the SFFS, followed by the US term spread, the consumer confidence index, and selling prices. Additional indicators from financial markets and tendency surveys are also included. Notably, the German term spread, a conventional early warning indicator, is omitted; instead, only the US term spread and the transatlantic term spread, i.e., the difference between the German and US term spreads, appear.

In contrast, all tree-based models select the unemployment rate as the leading indicator instead of energy prices. The Random Forest yields the most parsimonious indicator set, relying predominantly on well-documented financial market indicators such as the national term spread, the 10-year government bond yield, and overnight

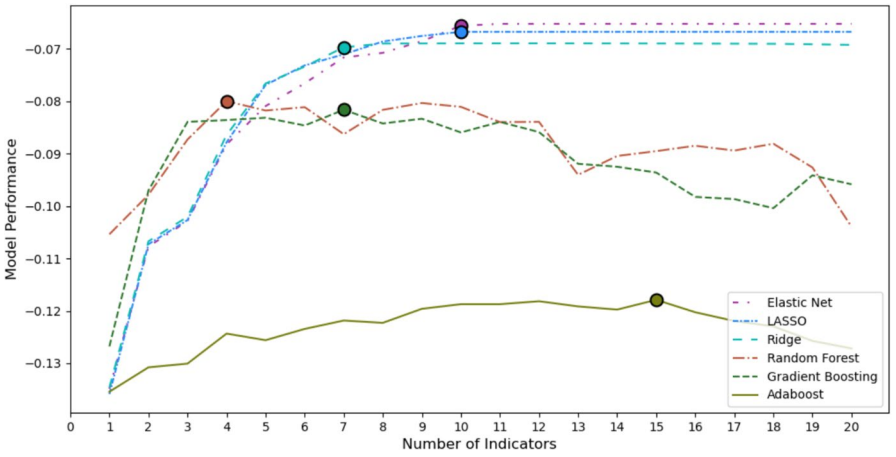


Fig. 2 Model performance after each iteration of the SFFS, measured as negative Brier-Score, points indicate how many features were selected

interbank rates. Furthermore, Random Forest exclusively employs static indicators and does not include any percentage changes or year-over-year differences.

AdaBoost diverges from both regression and Random Forest by not using any term spread-related indicator. It instead incorporates a unique set of price indicators, exchange rates, and share prices. In total, AdaBoost selects eight indicators not chosen by any other model. Gradient boosting, on the other hand, has more overlaps with the other models. Similar to the regression methods, it focuses on survey and financial market indicators and also selects the US term spread and the transatlantic spread.

Overall, the seven models exhibit considerable heterogeneity in both the number and nature of the selected indicators. Regression and Boosting rely on approximately twice as many predictors as Random Forest, with a stronger emphasis on survey-based expectations and energy prices. Across all models, energy prices, the US term spread, the transatlantic spread, the consumer confidence index, and order book indices emerge as the most frequently selected indicators, each appearing in four models.

5.2 Model Performance

Building on the selected features, this chapter evaluates the models' ability to predict recessions. Therefore, Fig. 3 illustrates the recession probabilities generated by the seven models against the periods where the models should predict a recession. In addition, the recession probabilities from a probit model with the term spread as the sole indicator are presented as a benchmark. A probability exceeding the horizontal line, which indicates a 50% threshold, leads to a recession prediction. The shaded areas correspond to recessions shifted 12 months backwards. These regions denote the periods during which a model should predict a recession.

The primary focus of the model evaluation lies on the forecasts generated for the hold-out sample, starting from the black line in 2006, as this data is fully unknown to the models. For the sake of completeness, the plots in Fig. 3 also show the recession

Table 1 Set of features selected for each classifier by SFFS. Ordered by category and frequency. Features denoted with "L" are leading features, that were picked in the first iteration of SFFS. The column "Count" shows how many classifiers have selected the feature

	LASSO	Ridge	Elastic Net	Random Forest	AdaBoost	Gradient Boosting	Count
Financial							
Term Spread USA (DF)	X	X	X			X	4
Term Spread Germany—USA (DF)	X	X	X			X	4
Call money/interbank rate (<24 h.)	X			X	X		3
BAA vs. 10-Year Treasury (DF)	X	X					2
Call money/interbank rate (<24 h.) (DF)					X		1
Long-term interest rates (10-year) (EURO)			X				1
Term Spread USA			X				1
Long-term interest rates (10-year)				X			1
Term Spread				X			1
Survey-based							
Consumer Confidence Indicator (SA)	X	X	X		X		4
Order Books (SA, DF)	X		X		X	X	4
Selling Prices (SA, DF)	X	X	X				3
Production Future Tendency (SA, DF)					X	X	2
Orders Inflow (SA)			X			X	2
Export Orders (SA)	X		X				2
Orders (SA, %)					X		1
Finished Goods Stocks (SA, DF)						X	1
Orders Inflow (SA, DF)	X						1
Business Confidence indicator (SA, DF)		X					1
Prices							
Energy Prices (%)	L	L	L		X		4
Oil Prices (%)					X		2
Overall Prices (%)					X		1
Housing Prices (%)					X		1
Producer Price Index in Manufacturing (%)					X		1
Labour Market							
Unemployment rate (SA)				L	L	L	3
Unemployment rate					X		1
International							
Real Effective Exchange Rates Eurozone (%)					X		1
Stock Market							
OECD Share Prices Index (%)					X		1
Sum	10	7	10	4	15	7	

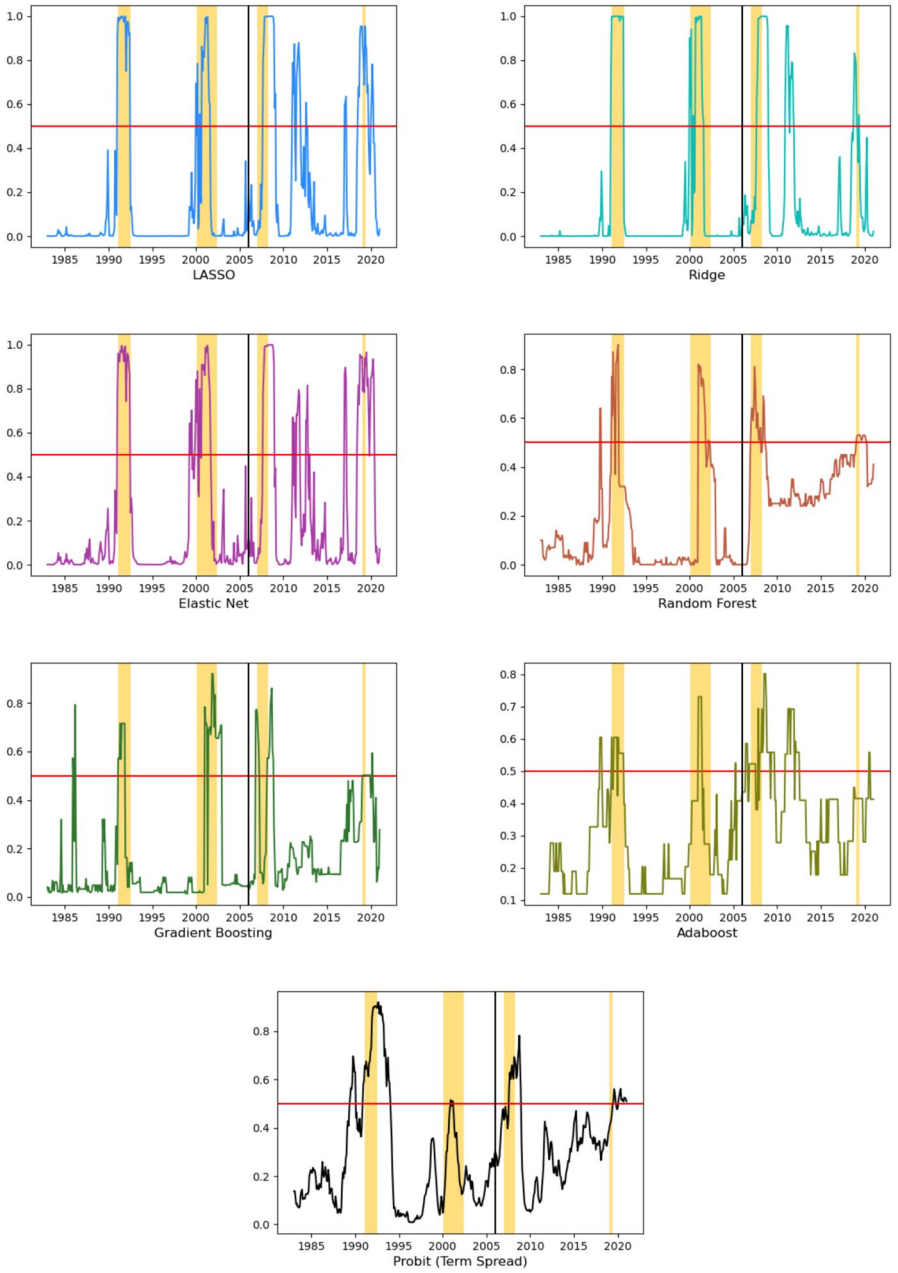


Fig. 3 Recession Probability Plots

probabilities derived from an expanding-window procedure for the period before 2006. For each validation fold, the model—equipped with the features and hyperparameters selected during validation—is trained on all data up to the fold's start date, then evaluated on that fold. Consequently, forecasts for January 1983–December

1991 originate from a model trained on data before January 1983, while forecasts for January 1992–January 2001 come from a model trained until December 1991. However, the recession probabilities prior to 2006 may exhibit an overly optimistic bias, as feature selection and hyperparameter tuning were conducted on this part of data.

When looking at the plots, one can see how well the model predictions match with the actual state of the economy. Most models predict the true recessions during the periods highlighted; only AdaBoost and the Probit model fail to predict the 2019 recession. Both Random Forest and Gradient Boosting barely crossed the 50% threshold in forecast probability, indicating a very tight prediction.

During most of the expansion phases, the models predict expansions. There is one major exception: LASSO, Ridge and Elastic Net predict a fictitious recession between 2010 and 2012, which mainly resulted from the increasing selling and energy prices. It should be noted that one year after the recession forecasts, GDP growth rates were very low (below 1%) and even negative in the first quarter of 2013. Although this period is not defined as a recession by the GCEE, it is close to a technical recession. Heilemann (2019) discusses Germany's various recession phases and refers to the period from April 2012 to April 2013 as a recession (Heilemann, 2019, p. 547). The Council of Economic Experts, whose definition is used, also discussed this period being a recession, but it didn't become one by a narrow margin. (Breuer et al., 2018, p. 31). The high recession probabilities are therefore not a bad prediction. Moreover, it shows that the models are very sensitive and already give recession signals when the GDP growth rates are small and negative but not yet long enough for a recession. Secondly, these predictions once again highlight the problem that definitions of recession can contain subjective components, but models, on the other hand, are strictly based on data.

Table 2 presents classifier performance metrics for the validation folds (left columns) and the hold-out sample (right columns). To evaluate the performance of the SFFS models, two benchmark models are shown for each classifier. One model with all features included ("ALL") and another model in which the ten best features from the full model are selected ("SFM"). The ten best features were selected in the regressions based on the coefficient and in the tree-based methods based on their mean decrease in impurity. The final three rows report additional benchmarks: a logistic regression and a decision tree without any feature selection and a probit model including only the term spread.

Models that restrict features—whether through SFFS or SFM—outperform full models without feature selection in the validation folds (left columns), yielding higher metrics and, in some cases, more timely forecasts. For example, applying feature selection to Random Forest cuts its average forecast delay from 11 to 6 months. A comparison of all models with SFFS shows that the regression models can adapt better to the data than the tree-based methods. Their higher goodness of fit became already evident in Fig. 1, where the Brier score for the regression models was slightly higher.

Out of sample, the tree-based SFFS models have an improved forecasting power compared to the full models and are also better than the SFM models. SFFS increases all metrics and reduces the delay. For example, from 10 to 2 for Random Forest

Table 2 In- and Out-Of-Sample Performance

	Validation Sample 1983/01 – 2005/12						Out-Of-Sample 2006/01 – 2020/12					
	AUC	REC	PRE	F1	BAC	DL	AUC	REC	PRE	F1	BAC	DL
LASSO												
ALL	0.72	0.52	0.25	0.34	0.59	0	0.89	1.00	0.29	0.45	0.79	0
SFM	0.85	0.71	0.41	0.52	0.75	**0**	0.90	1.00	0.29	0.45	0.79	0
SFFS	**0.94**	**0.71**	**1.00**	**0.83**	**0.85**	**0**	**0.84**	**0.78**	**0.27**	**0.40**	**0.71**	**2**
Ridge												
ALL	0.63	0.40	0.33	0.36	0.61	6	0.90	1.00	0.40	0.57	0.87	0
SFM	0.78	0.50	0.49	0.49	0.70	**6**	0.83	0.70	0.31	0.43	0.72	4
SFFS	**0.94**	**0.71**	**0.94**	**0.81**	**0.85**	**0**	**0.87**	**0.74**	**0.42**	**0.53**	**0.78**	**2**
ENET												
ALL	0.68	0.50	0.26	0.34	0.60	4	0.90	1.00	0.30	0.46	0.80	0
SFM	0.83	0.65	0.45	0.53	0.74	0	0.83	0.70	0.26	0.38	0.68	4
SFFS	**0.94**	**0.77**	**0.80**	**0.79**	**0.87**	**0**	**0.84**	**0.78**	**0.31**	**0.44**	**0.74**	**2**
RF												
ALL	0.73	0.17	0.44	0.24	0.56	11	0.81	0.30	0.42	0.35	0.61	10
SFM	0.78	0.29	0.40	0.34	0.60	6	0.78	0.26	0.25	0.25	0.56	10
SFFS	**0.81**	**0.44**	**0.91**	**0.59**	**0.71**	**6**	**0.96**	**0.78**	**0.66**	**0.71**	**0.85**	**2**
GB												
ALL	0.56	0.13	0.15	0.13	0.49	6	0.56	0.07	0.13	0.09	0.49	/
SFM	0.82	0.77	0.55	0.64	0.82	0	0.69	0.22	0.21	0.21	0.54	10
SFFS	**0.75**	**0.56**	**0.77**	**0.65**	**0.76**	**6**	**0.73**	**0.44**	**0.46**	**0.45**	**0.68**	**2**
ADAB												
ALL	0.73	0.23	0.44	0.30	0.58	6	0.61	0.26	0.20	0.23	0.54	10
SFM	0.79	0.33	0.39	0.36	0.61	6	0.65	0.26	0.19	0.22	0.53	10
SFFS	**0.89**	**0.42**	**0.83**	**0.56**	**0.70**	**6**	**0.71**	**0.52**	**0.26**	**0.35**	**0.63**	**0**
Logit	0.67	0.48	0.31	0.38	0.63	4	0.90	1.00	0.33	0.50	0.82	0
Probit	0.78	0.42	0.42	0.42	0.65	5	0.84	0.41	0.33	0.37	0.63	8
Tree	0.50	0.27	0.17	0.21	0.49	6	0.51	0.26	0.21	0.23	0.55	10

and from 10 to 0 for AdaBoost. Nevertheless, aside from Random Forest, tree-based models still trail regression models in predictive power.

In the case of regression models, SFFS likewise yields high-performing models; however, compared with the full models, it lowers performance metrics by approximately 10% and introduces a modest delay. With the exception of LASSO, SFFS consistently outperforms coefficient-based feature selection (SFM) across all model classes.

The fact that linear models without feature selection achieve stable results both in-sample and out-of-sample suggests that the relationship between the indicators and the recessions can be modelled well with coefficients. The weaker performance of LASSO—when compared with standard logistic regression, Ridge, and Elastic Net—suggest that the removal of less relevant predictors may harm forecasting power. In contrast, Ridge and Elastic Net shrink the coefficients toward zero rather than removing them entirely, preserving the marginal signals and therefore predictive power.

To better understand the out-of-sample results, Table 3 decomposes the out-of-sample period into the financial crisis (2006–2015) and the period of quantitative easing leading into the recession 2019 (2015–2019). During the financial crisis, regression models exhibit a clear performance drop when SFFS is applied. However, in the 2019 downturn, the same models match or even outperform the benchmark. Tree-based methods tell a different story: Feature selection with SFFS consistently increased their predictive power across both episodes. Without feature selection Random Forest and Gradient Boosting cannot detect the 2019 recession.

The 2008 recession can be forecasted very well by the probit model. From the SFFS models, only Random Forest can improve the term spread prediction. There are two reasons for this: First, the 2008 recession was caused by the financial crisis. Models that rely exclusively on financial market indicators have a clear advantage against models with more diversification regarding their feature set. Secondly, the poor performance of the SFFS regression models stems not only from their pre-recession predictions in 2008 but also from their prediction of a fictitious recession in 2010. Although Ridge recognizes just as many months of the 2008 recession as the probit model (both models have the same recall), it has a lower precision, F1 and AUC due to the misclassified expansion phase in 2010–2011.

For the 2019 recession, all models with SFFS performed better than the probit model. The latter loses a lot of predictive power due to quantitative easing. Probit and Random Forest, which rely solely on financial market indicators, now have a weaker performance than the more diversified models.

5.3 Structural Breaks

After evaluating model performance, attention turns to the time-varying importance of individual indicators and their contribution to recession forecasts. Of particular interest are structural breaks in the data, i.e., differences between the training and test data. For example, if indicators selected via SFFS were predictive during downturns in the train sample but have since lost forecasting power, this should be addressed methodically, and on the other side, when indicators gain importance.

Table 3 Out-Of-Sample Performance, decomposed for the periods 2006–2014 and 2015–2021

	Financial Crisis 2006/01 – 2014/12						Quantitative Easing / Recession 2019 2015/01 – 2020/12					
	AUC	REC	PRE	F1	BAC	DL	AUC	REC	PRE	F1	BAC	DL
LASSO												
ALL	0.83	1.00	0.29	0.45	0.76	0	0.99	1.00	0.30	0.46	0.84	0
SFM	0.84	1.00	0.29	0.45	0.76	0	0.98	1.00	0.29	0.45	0.83	0
SFFS	0.74	0.67	0.26	0.37	0.64	5	0.99	1.00	0.30	0.46	0.84	0
Ridge												
ALL	0.84	1.00	0.37	0.54	0.83	0	0.98	1.00	0.47	0.64	0.92	0
SFM	0.74	0.56	0.24	0.34	0.61	8	0.98	1.00	0.45	0.62	0.91	0
SFFS	0.81	0.61	0.34	0.44	0.69	7	0.98	1.00	0.56	0.72	0.95	0
ENET												
ALL	0.83	1.00	0.30	0.46	0.76	0	0.99	1.00	0.31	0.47	0.84	0
SFM	0.74	0.56	0.22	0.32	0.58	8	0.98	1.00	0.32	0.49	0.85	0
SFFS	0.77	0.67	0.32	0.43	0.69	5	0.96	1.00	0.30	0.46	0.84	0
RF												
ALL	0.77	0.44	0.42	0.43	0.66	10	0.92	0.00	0.00	0.00	0.50	/
SFM	0.76	0.39	0.25	0.30	0.58	10	0.77	0.00	0.00	0.00	0.50	/
SFFS	0.99	0.89	0.89	0.89	0.93	0	0.85	0.56	0.36	0.43	0.71	4
GB												
ALL	0.54	0.11	0.14	0.13	0.49	/	0.59	0.00	0.00	0.00	0.48	/
SFM	0.70	0.33	0.21	0.26	0.54	10	0.59	0.00	0.00	0.00	0.50	/
SFFS	0.68	0.39	0.50	0.44	0.66	0	0.85	0.56	0.42	0.48	0.72	4
ADAB												
ALL	0.57	0.39	0.21	0.27	0.54	10	0.66	0.00	0.00	0.00	0.49	/
SFM	0.62	0.39	0.21	0.27	0.54	10	0.67	0.00	0.00	0.00	0.48	/
SFFS	0.60	0.78	0.27	0.40	0.68	0	0.92	0.00	0.00	0.00	0.48	/
Logit	0.85	1.00	0.32	0.48	0.78	8	0.98	1.00	0.36	0.53	0.88	0
Probit	0.94	0.61	0.65	0.63	0.77	7	0.51	0.00	0.00	0.00	0.38	/
Tree	0.54	0.39	0.21	0.27	0.55	10	0.39	0.00	0.00	0.00	0.50	/

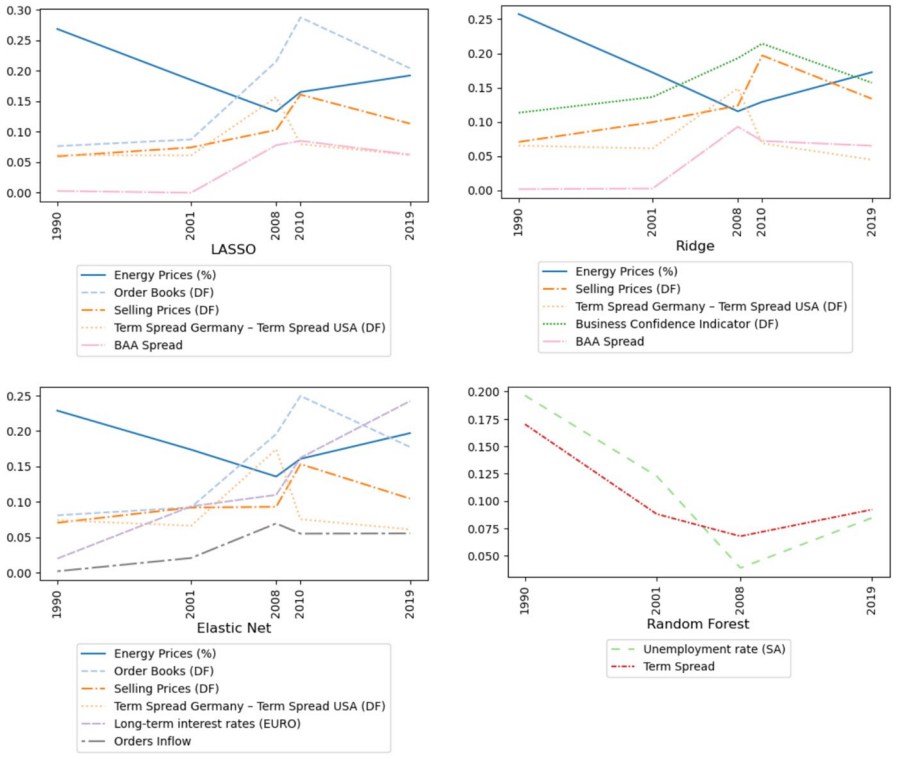


Fig. 4 Mean absolute SHAP values for indicators over time. The figure is restricted to indicators with a minimum mean absolute SHAP value of 0.05. The calculations were carried out for the following periods: 1983–1991, 1992–2001, 2006–2015 (only Random Forest), 2006–2009, 2009–2015, 2015–2020. (The plots for Gradient Boosting and AdaBoost can be found in "Mean absolute SHAP values" in the appendix. There are also plots showing the evolution of relevant indicators over time.)

To test, whether there are such structural breaks, SHAP values are computed for every indicator of the four SFFS models with a high performance.[8] Then the mean absolute SHAP value is calculated for each indicator on five different subperiods of the dataset.[9] The mean absolute SHAP value quantifies the strength of each indicator's impact on predictions, without indicating whether the indicator raises or lowers recession risk. Figure 4 plots the most important indicators according to mean absolute SHAP value and traces their evolution over time. For clarity, the plots omit any indicator with a mean absolute SHAP value below 0.05 over the whole sample is omitted.

It is important to emphasise that it is not the relationship between an indicator and the recession that is being examined, but rather the modelled connection. There-

[8] The plots for AdaBoost and Gradient Boosting can be found in the appendix „Mean absolute SHAP values ".

[9] The sub-periods are divided according to the recession phases, with the fictitious recession being included separately in the regression models.

fore, the SHAP trends can vary, especially between regression-based and tree-based models.

Across the linear regression models, two major trends can be observed: The survey-based indicators (selling prices, order books, business confidence indicator) have gained influence in the test data set. Energy prices, on the other hand, have lost importance in the training data, especially during the 2008 recession. A shift from energy prices to survey-based indicators can therefore be seen here, but all models show strong indicators across all periods. However, the model generates strong predictions in both samples, which can be explained by the fact that the survey-based indicators compensate for the weakness of energy prices. The term spread difference plays a constant role in forecasts and is only temporarily more relevant during the financial crisis.

In Random Forest, the two most important indicators, namely the term spread and the unemployment rate, lose much of their predictive power. Random Forest lacks strong predictors for the period after 2008, which also became visible in Fig. 3, where the model had an increased recession risk during the expansion periods and where only the recession of 2019 was predicted with recession risks only slightly above the threshold of 50%.

These changes can be explained as follows: Energy prices lose relative importance in the test sample because the training period encompasses two oil-crisis-driven recessions (1973 and 1979), whereas the 2001 and 2008 downturns were triggered by technology and financial-market disruptions rather than energy-market shocks. The visible changes in the impact of the term spread and the interest rates can be associated to quantitative easing. While a term spread close to zero was a strong recession signal in the training data, it is no longer after 2014. During Quantitative Easing, interest rates are so low that the term spread is consistently very small and close to zero. The structural break in the unemployment rate can be attributed to changes in the labour market environment, e.g. the increased use of short-time working during the financial crisis or the labour market reforms of Agenda 2010. To address the described structural breaks methodically and improve the predictive power of the models, an adjustment on the setting will be done, which is called dynamic setting and presented in the next section.

5.4 Dynamic Setting

In the dynamic setting, it is assumed that forecasters update the feature set and hyperparameters after each recession to incorporate the information from the most recent recession. Accordingly, feature selection, hyperparameter optimization, and model training are first undertaken following the 1993 downturn and then repeated after the 2001 and 2008 recessions. This expanding-window approach mitigates the risk of a lower model performance due to structural breaks by reassessing variable relevance and tuning parameters considering the newest recession. Moreover, to expand the test data, the 2001 recession is now included in the out-of-sample phase.

Table 4 shows the results summarised for the new out-of-sample period from 1999 to 2021. Regression models again outperform tree-based approaches, with LASSO delivering the strongest overall metrics. Applying SFFS to the regression mod-

Table 4 Out-Of-Sample Performance in the Dynamic Setting

	Out-Of-Sample 1999/01 – 2020/12								
	AUC	REC	PRE	F1	BAC	DL Av	DL 2001	DL 2008	DL 2019
LASSO									
ALL	0,80	0,84	0,35	0,50	0,71	0	0	0	0
SFM	0,85	0,88	0,37	0,52	0,74	0	0	0	0
SFFS	0,77	0,70	0,50	0,58	0,75	1	0	0	5
Ridge									
ALL	0,68	0,54	0,40	0,46	0,66	0	/	0	0
SFM	0,70	0,51	0,34	0,41	0,62	6	11	8	0
SFFS	0,69	0,44	0,51	0,47	0,66	6	10	7	1
ENET									
ALL	0,75	0,74	0,35	0,47	0,68	3	9	0	0
SFM	0,76	0,63	0,35	0,45	0,65	2	0	8	0
SFFS	0,77	0,68	0,43	0,53	0,72	1	0	0	5
RF									
ALL	0,57	0,14	0,44	0,21	0,55	10	/	10	/
SFM	0,61	0,12	0,25	0,16	0,51	10	/	10	/
SFFS	0,64	0,28	0,89	0,43	0,64	0	/	0	/
GB									
ALL	0,46	0,16	0,26	0,20	0,52	/	/	/	/
SFM	0,56	0,21	0,40	0,28	0,56	11	12	10	/
SFFS	0,45	0,28	0,48	0,36	0,60	0	/	0	0
ADAB									
ALL	0,69	0,12	0,32	0,18	0,53	10	/	10	/
SFM	0,65	0,21	0,50	0,30	0,58	11	11	12	/
SFFS	0,67	0,56	0,42	0,48	0,67	0	0	0	0
Logit	0,70	0,63	0,36	0,46	0,66	3	11	0	0
Probit	0,70	0,23	0,37	0,28	0,56	9	11	7	9
Tree	0,58	0,30	0,29	0,30	0,55	8	12	11	3

els boosts F1 score and balanced accuracy, but it comes at the expense of reduced recall—a trade-off that was also visible in the static setting. Compared to the static setting, however, the SFF models are not worse, but rather better. If the prediction of expansion and recession is equally important, the SFFS models should be used. However, if predicting all recession phases is paramount, the baseline models are usually the best choice because as they have a higher recall. The only exception is LASSO, where the recall is highest when using the SFM model.

No clear statement can be made regarding the timeliness of the forecasts. In LASSO, the forecasts are delayed by five months for the 2019 recession due to the SFFS. With Elastic Net, the delay is reduced from an average of three months to one month. By contrast, Ridge's delay worsens under SFFS because it flags an additional recession phase—albeit ten months late—that the baseline model ignored (denoted by "/" and not included in the average delay).

Regarding the tree-based methods, SFFS generally increases most metrics, yet they are still worse than the regression models. In contrast to the static setting, Ran-

dom Forest's performance worsens under SFFS: it fails to detect the 2001 recession and, after feature re-selection and hyperparameter tuning post-crisis, also misses the 2019 downturn. Although it assigns higher recession probabilities for 2019, none exceed the 50% decision threshold.

6 Discussion and Conclusion

The findings demonstrate that regularized regression models—LASSO, Ridge, and Elastic Net—are the most effective tools for forecasting recessions in Germany. Across both static and dynamic evaluations, these parametric models deliver consistently robust and reliable results and outperform the benchmark probit model.

For the regularized regression models, the impact of SFFS varies by recession period and setting. No consistent improvement or significant decrease in forecasting power is visible. When the primary goal is to predict every downturn (i.e., maximize recall), the baseline models without SFFS are the best choice. When a false recession signal is given the same weight as a false expansion signal (i.e., maximizing the F1 score), then the SFFS models are a good alternative.

In contrast, tree-based methods have proven to be less performant overall, especially when no feature selection has taken place. The internal mechanisms of Random Forest and Boosting designed to give relevant features more weight were not sufficient for the German dataset to build reliable models. Tree-based models using all indicators quickly suffer from overfitting, when they are applied out of sample. A more rigorous feature selection (either with SFFS or SFM) consistently improved the metrics of those classifiers but did not produce models that were better than the regression models. Random Forest performed superior to the regressions in the static setting but did not pass the robustness check of the dynamic approach.

Several factors may explain why tree-based models have a lower forecasting power. First, the relationships between indicators and recessions appear predominantly linear, inherently favouring regression-based approaches. Second, the sample size may be too small for random forests or boosting to fully leverage their nonlinear flexibility—a conclusion supported by their weaker validation performance relative to linear models. Finally, these methods' nonlinear search can emphasize predictors that, despite capturing complex patterns, contribute little true signal, leading to a less relevant feature set compared to the regression models. The high forecasting power of the regression models on the other side, shows that recessions in Germany can be forecasted well in a linear way. Beyond forecasting power, regression models also have the advantage that they are easy to interpret through their coefficient estimates.

Overall, many findings from Germany are consistent with conclusions that have already been obtained from research on US recessions. For example, the high forecasting power of regularized regressions and poor performance of Random Forest and Boosting was also pointed out in a similar research setting with one-year-ahead forecasts for the USA (Vrontos et al., 2021, p. 660). The small indicator sets, in general, support the results from Ng (2014) and Puglia and Tucker (2020), who have already pointed out that a few strong indicators are sufficient for forecasting.

Many indicators, selected in SFFS, have already been used frequently in the literature (c.f. Nissilä, 2020; Vrontos et al., 2021). For example, Random Forest selected very conventional indicators such as the term spread and short-term interest rates and the regression models further employ many survey-based indicators, the importance of which has often been emphasized. (Christiansen et al., 2014).

A very central observation, however, is, that the best-performing models, i.e., the regression models, rely strongly on the US indicators for their forecasts. They use the US term spread, and the transatlantic term spread, but not the German term spread. In addition, LASSO and Ridge also include the BAA spread. The indicator set therefore, displays that the US business cycles have a strong relevance to the German ones.

The high forecasting power of the US financial indicators is notable because models in the literature often choose the national term spreads as starting points. But although the US and transatlantic term spreads were included in the models, they were not the starting points for the models. When building a model with only one indicator, the growth rate of the energy prices was the most powerful indicator in the regression models, which is striking because this price index has not been frequently mentioned as so powerful in the literature.

Conversely, it is noteworthy that neither the DAX nor the SP500 was selected, although the last one frequently appeared in the literature. There are two possible reasons for that: First, the DAX comprises only 30 companies and thus does not represent the German economy as detailed as the SP500 does for the US. Second, the stock market indices might perform better at a shorter forecast horizon, that is, less than one year in advance (Nissilä, 2020, p. 23–25).

For the application on German business cycles, SFFS was able to produce performant forecasting models while revealing which indicators matter most. However, selecting an optimal feature set is NP-hard as the number of possible feature subsets grows exponentially with dimensionality, so practical methods rely on assumptions or heuristics to search that space efficiently. SFFS is just one algorithm among many; new methods for feature selection appear constantly. Because of that and the fact that feature-selection methods have seen limited application in recession forecasting, considerable scope remains for future research that evaluates their performance across countries and historical periods.

7 Appendix

List of Indicators

Subject	L	%	Δ	Source	OECD / FRED Code
Financial Markets					
Long-term interest rates (10-year) (EURO)	1	0	1	OECD	EA19_IRLTLT01_ST
Monetary Aggregate M3	0	1	0	OECD	EA19_MABMM301_IXOB
Monetary Aggregate M3 (SA)	0	1	0	OECD	EA19_MABMM301_GYSA
Monetary Aggregate (M1)	0	1	0	OECD	EA19_MANMM101_IXOB
Monetary Aggregate (M1) (SA)	0	1	0	OECD	EA19_MANMM101_IXOBSA
3-month or 90-day rates and yields	1	0	1	OECD	IR3TIB01_ST

Subject	L	%	Δ	Source	OECD / FRED Code
Long-term interest rates (10-years)	1	0	1	OECD	IRLTLT01_ST
Call money/interbank rate (<24 h.)	1	0	1	OECD	IRSTCI01_ST
Term Spread	1	0	1	OECD	LOCOSIOR_ST
Moody's Seasoned Aaa Corporate Bond Yield Relative to Yield on 10-Year Treasury Constant Maturity	1	0	1	FRED	AAA10YM
Moody's Seasoned Aaa Corporate Bond Minus Federal Funds Rate	1	0	1	FRED	AAAFFM
Moody's Seasoned Baa Corporate Bond Yield Relative to Yield on 10-Year Treasury Constant Maturity	1	0	1	FRED	BAA10YM
Moody's Seasoned Baa Corporate Bond Minus Federal Funds Rate	1	0	1	FRED	BAAFFM
Term Spread USA	1	0	1	FRED	TB6SMFFM
Term Spread USA (Version 2)	1	0	1	FRED	T10YFFM
Term Spread Germany—Term Spread USA	1	0	1		own calculation

Subject	L	%	Δ	Source	OECD / FRED Code
International					
Real Effective Exchange Rates	0	1	0	OECD	CCRETT01_IXOB
Real Effective Exchange Rates (EURO)	0	1	0	OECD	EA19_CCRETT01_IXOB
US$ exchange rate (National currency:USD)	1	0	1	OECD	CCUSMA02_ST
US$ exchange rate (USD:national currency)	1	0	1	OECD	CCUSSP01_ST
Export Orders	1	0	1	OECD	LOCOBXOR_STSA
Labor Market					
Job vacancies	0	1	0	OECD	LMJVTTUV_ST
Job vacancies (SA)	0	1	0	OECD	LMJVTTUV_STSA
Registered unemployment	0	0	1	OECD	LMUNRLTT_ST
Registered unemployment (SA)	0	0	1	OECD	LMUNRLTT_STSA
Unemployment rate	1	0	0	OECD	LMUNRRTT_ST
Unemployment rate (SA)	1	0	0	OECD	LMUNRRTT_STSA
Prices					
Food and non-Alcoholic beverages	0	1	0	OECD	CP010000_GY
Electricity, gas and other fuels	0	1	0	OECD	CP040500_GY
Overall	0	1	0	OECD	CPALTT01_GY
Energy	0	1	0	OECD	CPGREN01_GY
Housing	0	1	0	OECD	CPGRHO01_GY
All items non-food non-energy	0	1	0	OECD	CPGRLE01_GY
Oil Prices	0	1	0	FRED	WTISPLC

Subject	L	%	Δ	Source	OECD / FRED Code
Real Economy					
Orders	0	1	0	OECD	LOCOODOR_IXOBSA
Orders (Manufacturing)	0	1	0	OECD	ODMNTO01_IXOB
Producer Price Index (Manufacturing)	0	1	0	OECD	PIEAMP02_GY
Stock Market					
DAX	0	1	0	Yahoo Finance / Stehle	

Subject	L	%	Δ	Source	OECD / FRED Code
Standard & Poors 500	0	1	0	Shiller	
OECD Share Prices Index	0	1	0	OECD	SPASTT01_GY
Business Tendency / Consumer Confidence Surveys					
Business Situation	1	0	1	OECD	BSBUCT02_STSA
Business Confidence indicator	1	0	1	OECD	BSCICP02_STSA
Finished Goods Stocks	1	0	1	OECD	BSFGLV02_STSA
Order Books	1	0	1	OECD	BSOBLV02_STSA
Orders Inflow	1	0	1	OECD	BSOITE02_STSA
Production Future Tendency	1	0	1	OECD	BSPRFT02_STSA
Production Tendency	1	0	1	OECD	BSPRTE02_STSA
Selling Prices	1	0	1	OECD	BSSPFT02_STSA
Consumer Confidence Indicator	1	0	1	OECD	CSCICP02_STSA

The "OECD / FRED Code" column shows the official abbreviation of the indicator, which is used in the database
The columns "L", "%", " Δ " show whether an indicator is used in the analysis in level, as a percentage change or difference to the previous year. If the OECD code ends in "GY", the original variable was already available as a percentage change

Data Sources:
OECD: https://www.oecd-ilibrary.org/economics/data/main-economic-indicators_mei-data-en
FRED: https://fred.stlouisfed.org/
Yahoo Finance: https://de.finance.yahoo.com/quote/%5EGDAXI/
Stehle: https://www.econstor.eu/handle/10419/66277
Shiller: http://www.econ.yale.edu/~shiller/data.htm

Hyperparameters and Search Spaces

	Tuned Hyperparameters with Search Ranges and Fixed Hyperparameters	Fixed Hyperparameters of Classifier used in SFFS
Random Forests (sklearn.ensemble RandomForestClassifier)	max_depth ∈(1, 30, step = 1) max_samples ∈ [0.5, 1] max_features ∈ [0.5, "sqrt"] n_estimators = 100 class_weight = "balanced"	max_depth = 3 n_estimators = 100 class_weight = "balanced"
Ridge (sklearn.linear_model LogisticRegression)	C ∈(0.0001, 100, log = True) class_weight = "balanced" penalty = "l2"	C = 1 class_weight = "balanced" penalty = "l2"
LASSO (sklearn.linear_model LogisticRegression)	C ∈(0.0001, 100, log = True) class_weight = "balanced" penalty = "l1" solver = "liblinear"	C = 1 class_weight = "balanced" penalty = "l1" solver = "liblinear"
Elastic Net (sklearn.linear_model LogisticRegression)	C ∈ (0.0001, 100, log = True) L1_ratio ∈(0.1, 0.9, step = 0.1) class_weight = "balanced" penalty = "elasticnet" solver = "saga"	C = 1 l1_ratio = 0.5 class_weight = "balanced" penalty = "elasticnet" solver = "saga"
AdaBoost* (sklearn.ensemble AdaBoostClassifier)	n_estimators ∈ (1, 100) learning_rate ∈ (0.001, 1, log = True)	algorithm = "SAMME"
Gradient Boosting* (sklearn.ensemble GradientBoostingClassifier)	n_estimators ∈(1, 100) learning_rate ∈ (0.001, 0.1, log = True) max_depth ∈ (3, 10) min_impurity_decrease ∈ (1e-8, 0.05, log = True)	

	Tuned Hyperparameters with Search Ranges and Fixed Hyperparameters	Fixed Hyperparameters of Classifier used in SFFS
Classification Tree** (sklearn.tree. DecisionTreeClassifier)	max_depth ∈ (1, 10) min_samples_leaf ∈ (6, 12) min_impurity_decrease ∈ (1e-8, 0.05, log=True) class_weight="balanced"	
Logistic Regession** (sklearn.linear_model LogisticRegression)	penalty=None class_weight="balanced"	

* Boosting classifiers do not have the build-in hyperparameter "class_weight". To achieve the desired weighting, sample weights are passed on to the classifier via the fit() command

** Classification Tree and Logistic Regression are not used in the SFFS, therefore, there are no fixed parameters in the column

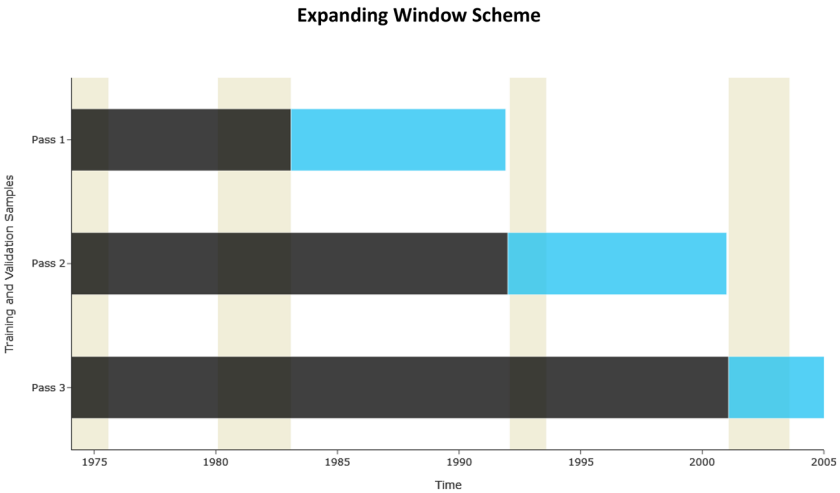
Tuned Hyperparameters

Classifier	Hyperparameter	ALL	SFFS	SFM
Train End: 1998–12				
LASSO	C	0,0078282	1,0920165	0,02947356
RIDGE	C	0,00020274	2,7,224,302	0,00052434
ENET	C	0,00109653	99,9,971,109	0,00171873
	l1_ratio	0,1	0,1	0,1
RF	max_depth	30	4	13
	max_samples	0,5	0,5	0,5
	max_features	"sqrt"	"sqrt"	"sqrt"
GB	n_estimators	14	60	64
	learning_rate	0,09108795	0,06953093	0,05788348
	max_depth	9	3	3
	min_impurity_decrease	1,2271E-06	0,02259611	0,49,881,557
ADA	n_estimators	33	35	14
	learning_rate	0,31,502,447	0,00138782	0,35,547,025
Tree	max_depth	8		
	min_samples_leaf	6		
	min_impurity_decrease	0,000554		

Classifier	Hyperparameter	ALL	SFFS	SFM
Train End: 2005–12 =Static Setting				
LASSO	C	0.00780392	1.20973832	0.028477678
RIDGE	C	0.00025215	5.63078309	0.00037516
ENET	C	0.00120857	96.4132091	0.00349446
	l1_ratio	0.1	0.2	0.1
RF	max_depth	2	11	2
	max_samples	0.5	0.5	0.5
	max_features	"sqrt"	0.5	"sqrt"
GB	n_estimators	39	89	61
	learning_rate	0.03375247	0.04695057	0.06448656
	max_depth	7	3	3
	min_impurity_decrease	1.0576E-08	5.5468E-08	0.46454994
ADAB	n_estimators	15	7	46
	learning_rate	0.57758658	0.99424787	0.57444066
Tree	max_depth	8		
	min_samples_leaf	6		

Classifier	Hyperparameter	ALL	SFFS	SFM
	min_impurity_decrease	0.00586183		

Classifier	Hyperparameter	ALL	SFFS	SFM
2010–12				
LASSO	C	0,00778052	1,27,890,973	0,00846555
RIDGE	C	0,00024234	3,18,365,423	0,00076828
ENET	C	0,00125064	99,8,596,028	0,00541734
	l1_ratio	0,1	0,1	0,8
RF	max_depth	2	6	2
	max_samples	0,5	0,5	0,5
	max_features	"sqrt"	0,5	"sqrt"
GB	n_estimators	39	91	20
	learning_rate	0,03096974	0,04073621	0,06693034
	max_depth	20	3	4
	min_impurity_decrease	0,00873277	7,3815E-07	4,5949E-06
ADAB	n_estimators	48	67	20
	learning_rate	0,29,314,254	0,99,348,073	0,62,297,614
Tree	max_depth	6		
	min_samples_leaf	7		
	min_impurity_decrease	3,05E-08		

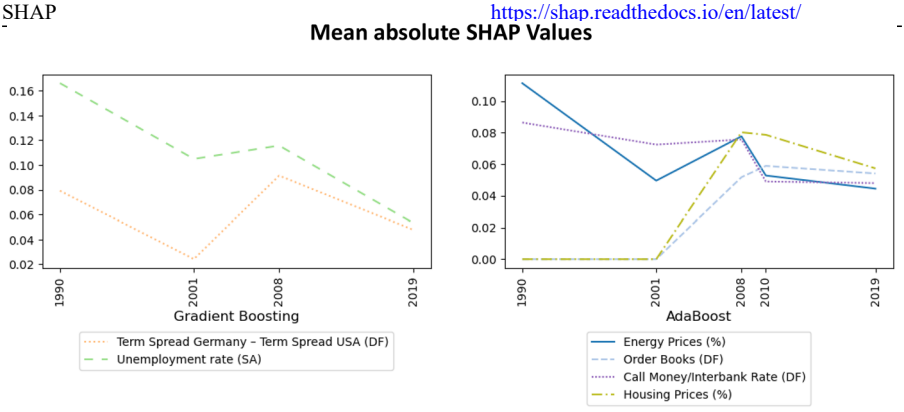


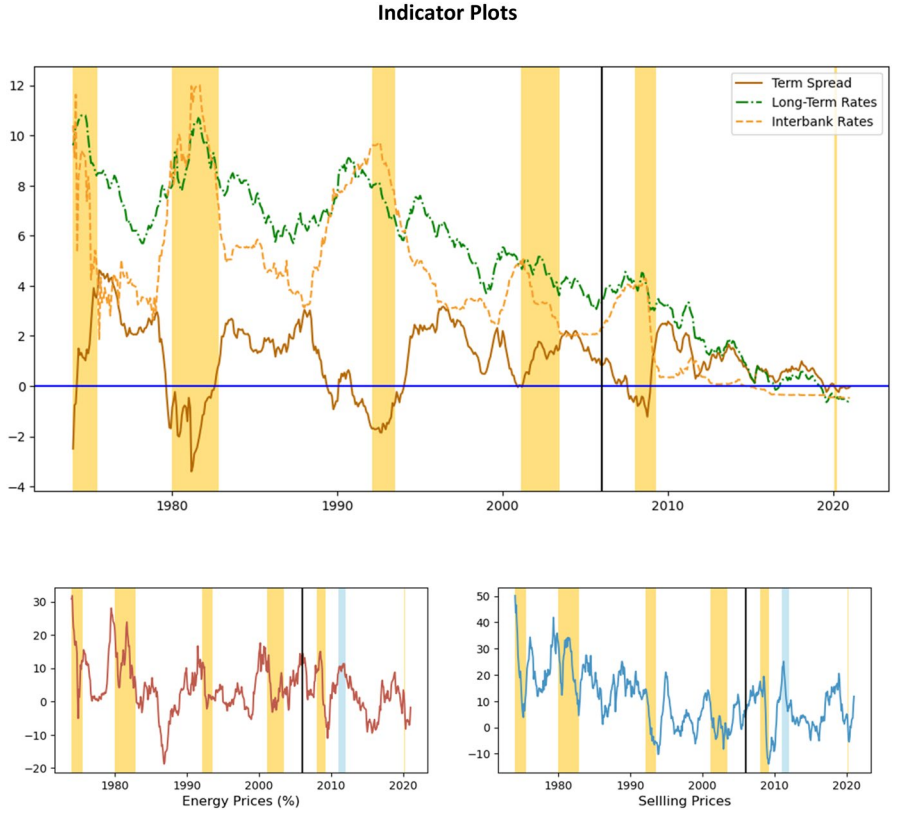
In the expanding window scheme, each validation fold (blue) contains the 12 months before the next recession period, but not the recession itself.

List of Python Packages

Python Package	Webpage
Logitboost	https://logitboost.readthedocs.io/
Matplotlib	https://matplotlib.org/
Mlxtend	https://rasbt.github.io/mlxtend/
Numpy	https://numpy.org/

Python Package	Webpage
Optuna	https://optuna.org/
Pandas	https://pandas.pydata.org/
Scikit-Learn (Sklearn)	https://scikit-learn.org/stable/
Statmodels	https://www.statsmodels.org/stable/index.html
SHAP	https://shap.readthedocs.io/en/latest/





Funding Open Access funding enabled and organized by Projekt DEAL.

Declarations

Conflict of interest The data that support the findings of this study are openly available. The data sources are listed in the appendix. This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors. The author declares that there is no Conflict of interest.

References

Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. (2019). Optuna: A next-generation hyperparameter optimization framework. In: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining, 2623–2631.

Berge, T. J. (2015). Predicting recessions with leading indicators: Model averaging and selection over the business cycle. *Journal of Forecasting, 34*(6), 455–471. https://doi.org/10.1002/for.2345

Bergstra, J., Bardenet, R., Bengio, Y., & Kégl, B. (2011). Algorithms for hyper-parameter optimization. Advances in neural information processing systems, 24.

Breiman, L. (2001). Random forest. *Machine Learning, 45*, 5–32. https://doi.org/10.1023/A:1010933404324

Breuer, S., Elstner, S., Kirsch, F., & Wieland, V. (2018). Datierung der deutschen Konjunkturzyklen–die Methode des Sachverständigenrates. *Arbeitspapier, 13*, 2018.

Cadahia Delgado, P., Congregado, E., Golpe, A., Vides, J. C. (2022). The Yield Curve as a Recession Leading Indicator. An Application for Gradient Boosting and Random Forest, International Journal of Interactive Multimedia and Artificial Intelligence, Special Issue, 7–19. https://doi.org/10.9781/ijimai.2022.02.006

Castle, J. L., Clements, M. P., & Hendry, D. F. (2015). Robust approaches to forecasting. *International Journal of Forecasting, 31*(1), 99–112. https://doi.org/10.1016/j.ijforecast.2014.11.002

Chauvet, M., & Potter, S. (2005). Forecasting recessions using the yield curve. *Journal of Forecasting, 24*(2), 77–103. https://doi.org/10.1002/for.932

Chen, Z., Iqbal, A., & Lai, H. (2011). Forecasting the probability of US recessions: A probit and dynamic factor modelling approach. *Canadian Journal of Economics, 44*(2), 651–672. https://doi.org/10.1111/j.1540-5982.2011.01648.x

Chinn, M., & Kucko, K. (2015). The predictive power of the yield curve across countries and time. *International Finance, 18*(2), 129–156. https://doi.org/10.1111/infi.12064

Choi, J., Ge, D., Kang, K. H., & Sohn, S. (2023). Yield spread selection in predicting recession probabilities. *Journal of Forecasting, 42*(7), 1772–1785. https://doi.org/10.1002/for.2980

Christiansen, C., Eriksen, J. N., & Møller, S. V. (2014). Forecasting US recessions: The role of sentiment. *Journal of Banking & Finance, 49*, 459–468. https://doi.org/10.1016/j.jbankfin.2014.06.017

Döpke, J., Fritsche, U., & Pierdzioch, C. (2015). Predicting Recessions in Germany With Boosted Regression Trees, DEP (Socioeconomics) Discussion Papers Macroeconomics and Finance Series, 5/2015.

Döpke, J., Fritsche, U., & Pierdzioch, C. (2017). Predicting recessions with boosted regression trees. *International Journal of Forecasting, 33*(4), 745–759. https://doi.org/10.1016/j.ijforecast.2017.02.003

Estrella, A., & Hardouvelis, G. A. (1991). The term structure as a predictor of real economic activity. The Journal of Finance, Vol. 46. *Nr., 2*, 555–576. https://doi.org/10.1111/j.1540-6261.1991.tb02674.x

Estrella, A., & Mishkin, F. S. (1998). Predicting U.S. recessions: Financial variables as leading indicators. *The Review of Economics and Statistics, 80*(1), 45–61. https://doi.org/10.1162/003465398557320

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*. https://doi.org/10.1214/aos/1013203451

GCEC (2023), Konjunkturzyklus-Datierung, https://www.sachverstaendigenrat-wirtschaft.de/themen/konjunktur-und-wachstum/konjunkturzyklus-datierung.html

Gilchrist, S., & Zakrajšek, E. (2012). Credit spreads and business cycle fluctuations. *American Economic Review, 102*(4), 1692–1720. https://doi.org/10.1257/aer.102.4.1692

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research, 3*, 1157–1182.

Hasse, J. B., & Lajaunie, Q. (2020). Does the yield curve signal recessions? New evidence from an international panel data analysis. AMSE Working Paper, Nr. 13.

Hastie, T., Tibshirani, R., Friedman, J. H., & Friedman, J. H. (2009). *The elements of statistical learning: Data mining, inference, and prediction*. Springer.

Heilemann, U. (2019). Rezessionen in der Bundesrepublik Deutschland von 1966 bis 2013. *Wirtschaftsdienst, 99*(8), 546–552. https://doi.org/10.1007/s10273-019-2489-6

Hendry, D. F. (2006). Robustifying forecasts from equilibrium-correction systems. *Journal of Econometrics, 135*(1–2), 399–426. https://doi.org/10.1016/j.jeconom.2005.07.029

Hoerl, A., & Kennard, R. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics, 12*(1), 55–67. https://doi.org/10.1080/00401706.1970.10488634

Jain, A., & Zongker, D. (1997). Feature selection: Evaluation, application, and small sample performance. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 19*(2), 153–158. https://doi.org/10.1109/34.574797

Jardet, C., & Meunier, B. (2022). Nowcasting world GDP growth with high-frequency data. *Journal of Forecasting, 41*(6), 1181–1200. https://doi.org/10.1002/for.2858

Kant, D., Pick, A., & Winter, J. D. (2025). Nowcasting GDP using machine learning methods. *Advances in Statistical Analysis, 109*(1), 1–24. https://doi.org/10.1007/s10182-024-00515-0

Li, M., Sun, H., Huang, Y., & Chen, H. (2024). Shapley value: From cooperative game to explainable artificial intelligence. *Autonomous Intelligent Systems, 4*(1), 2. https://doi.org/10.1007/s43684-023-00060-8

Lundberg, S. M., & Lee, S. I. (2017). A Unified Approach to Interpreting Model Predictions. Advances in Neural Information Processing Systems, 30.

NBER (2008). Business Cycle Dating Committee Announcement January 7, 2008. https://www.nber.org/news/business-cycle-dating-committee-announcement-january-7-2008

Nevasalmi, L. (2022). Recession forecasting with high-dimensional data. *Journal of Forecasting, 31*(4), 752–764. https://doi.org/10.1002/for.2823

Ng, S. (2014). Boosting recessions. *Canadian Journal of Economics, 47*(1), 1–34.

Nissilä, W. (2020). Probit based time series models in recession forecasting–A survey with an empirical illustration for Finland. Bank of Finland Economic Review, 7/2020.

Nyberg, H. (2010). Dynamic probit models and financial variables in recession forecasting. *Journal of Forecasting, 29*(1), 215–230. https://doi.org/10.1002/for.1161

Owen, G. (1977). Values of games with a priori unions. In: Mathematical economics and game theory: Essays in honor of Oskar Morgenstern (pp. 76–88). Berlin, Heidelberg: Springer Berlin Heidelberg.

Plakandaras, V., Cunado, J., Gupta, R., & Wohar, M. E. (2017). Do leading indicators forecast US recessions? A nonlinear re-evaluation using historical data. *International Finance, 20*(3), 289–316. https://doi.org/10.1111/infi.12111

Pönka, H. (2017). The role of credit in predicting US recessions. *Journal of Forecasting, 36*(5), 469–482. https://doi.org/10.1002/for.2448

Psimopoulos, A. (2020). Forecasting economic recessions using machine learning: An empirical study in six countries. *South-Eastern Europe Journal of Economics, 18*(1), 40–99.

Pudil, P., Novovičová, J., & Kittler, J. (1994). Floating search methods in feature selection. *Pattern Recognition Letters, 15*(11), 1119–1125. https://doi.org/10.1016/0167-8655(94)90127-9

Puglia, M. & Tucker, A. (2020). Machine Learning, the Treasury Yield Curve and Recession Forecasting, Finance and Economics Discussion Series 2020–038, Board of Governors of the Federal Reserve System (U.S.).

Puglia, M., & Tucker, A. (2021). Neural networks, the Treasury yield curve, and recession forecasting. *The Journal of Financial Data Science, 3*(2), 149–175. https://doi.org/10.3905/jfds.2021.1.061

Raschka, S., & Mirjalili, V. (2017). Python machine learning: machine learning and deep learning with Python, scikit-learn, and TensorFlow (Second edition). Birmingham, Packt.

Shapley, L. (1953). A Value for n-Person Games. In: Contributions to the Theory of Games II (1953) pp. 307–317.

Stehle, R., Maier, J., & Huber, R. (1996). Rückberechnung des DAX für die Jahre 1955 bis 1987. SFB 373 Discussion Paper 7.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological), 58*(1), 267–288. https://doi.org/10.1111/j.2517-6161.1996.tb02080.x

Vrontos, S. D., Galakis, J., & Vrontos, I. D. (2021). Modeling and predicting US recessions using machine learning techniques. *International Journal of Forecasting, 37*(2), 647–671. https://doi.org/10.1016/j.ijforecast.2020.08.005

Wang, S., Tang, J., & Liu, H. (2017). Feature selection. In Encyclopedia of Machine Learning and Data Mining (pp. 503–511). Springer, Boston, MA. https://doi.org/10.1007/978-1-4899-7687-1_101

Yazdani, A. (2020). Machine learning prediction of recessions: An imbalanced classification approach. *The Journal of Financial Data Science, 2*(4), 21–32. https://doi.org/10.3905/jfds.2020.1.040

Zhao, Z., Anand, R., & Wang, M. (2019). Maximum relevance and minimum redundancy feature selection methods for a marketing machine learning platform. In: 2019 IEEE international conference on data science and advanced analytics (DSAA) (pp. 442–452). IEEE. https://doi.org/10.1109/DSAA.2019.00059

Zou, H., & 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. https://doi.org/10.1111/j.1467-9868.2005.00503.x

Zyatkov, N., & Krivorotko, O. (2021). Forecasting Recessions in the US Economy Using Machine Learning Methods. 2021 17th International Asian School-Seminar" Optimization Problems of Complex Systems (OPCS), 139–146. https://doi.org/10.1109/OPCS53376.2021.9588678