Erasmus University Rotterdam
Erasmus School of Economics

Bachelor Thesis Financial Econometrics

Janou van Well
493653

Predicting GDP growth for European countries using numerous dimensionality reduction techniques

Abstract

The importance of big data analysis has been demonstrated through its increased use in contemporary research. To efficiently process large data sets, researchers apply numerous dimensionality reduction techniques. In this paper, several of these techniques are put into use to investigate the predictability of the growth in GDP of 34 European countries. Both principal component analysis and independent component analysis are applied as techniques to reduce the dimensionality of the data, while preserving the information contained in the data set. On top of these factor estimation methods, a set of machine learning, variable selection and shrinkage methods are evaluated: boosting, ridge regression, least angle regression, elastic net and non-negative garotte. The predictions generated by these methods are then compared to those of a number of benchmark models. In general, the results show that the factor estimation models outperform the benchmark models and can successfully be used to predict GDP growth. The larger number of observations used in the recursive estimation method leads to improvements of the results for this forecasting method when compared to the rolling window method. On top of this, it is found that the boosting algorithm and ridge regression generate the most accurate predictions.

Supervisor: Prof. dr. P.H. Franses
Second assessor: Dr. C. Cavicchia

July 2021

Contents

1 Introduction

'Big data' is a term that is used to describe large volumes of structured and unstructured data sets that are too large and too complex for traditional data processing applications. De Mauro et al. (2016) define big data as "the information asset characterised by such a high volume, velocity and variety to require specific technology and analytical methods for its transformation into value". The information that is captured in these large data sets can be of great value to organisations and companies. Proper big data analysis provides insights that may lead to enhanced decision making with benefits such as cost saving and time reduction. Today, these benefits are reaped by many industries, including the financial sector. One of the applications is the prediction in the growth of the gross domestic product (GDP) of a country. The GDP of a country is an accurate indicator of the size of the economy and its growth rate gives in indication of a country's economic growth. Knowing how the economy will develop could play an important role in the optimisation of appropriate monetary policy by central banks. Therefore, this paper looks into the prediction of the GDP of 34 European countries.

This article is focused around the work of Kim and Swanson (2018). In their paper, they analyse whether big data is useful for modelling low frequency macroeconomic variables. To do so, diffusion indices are constructed using three types of factor estimation methods: principal component analysis (PCA), independent component analysis (ICA) and sparse principal component analysis (SPCA). On top of this, they evaluate a set of machine learning methods: bagging, boosting, Bayesian model averaging, ridge regression, least angle regression, the elastic net and non-negative garotte. When forecasting their target variables, they consider a total of six specification types. Finally, they investigate both a recursive as well as a rolling data window method to forecast. After having obtained all of their forecasts, Kim and Swanson (2018) compare the different models by means of the Mean Square Forecast Error (MSFE) as well as the test statistic of Diebold and Mariano (2002). In this article, most of these methods and approaches are replicated. As for the estimation of the factors, this paper looks into PCA and ICA. The machine learning methods that are considered are boosting, the ridge regression, least angle regression, the elastic net and non-negative garotte. On top of this, there are three specification types included and both the recursive and the rolling data window methods are analysed. However, in this paper forecasts are obtained and evaluated for the GDP of European countries and thus a different data set is investigated.

The data set used by Kim and Swanson (2018) uses a total of 144 U.S. monthly macroeconomic time series over the period from January 1960 up to May 2009. For each of these variables, this adds up to a total of 593 observations. For 11 of the variables, there are forecasts constructed. It is noted that the number of variables is of the same order of magnitude as the number of observations. Keeping in mind these interesting properties of the data set of Kim and Swanson (2018), it is chosen in this paper to investigate and predict the GDP of European countries. This paper attempts to answer the question to what extent the growth of GDP can be predicted for European countries and what differences there are among this predictability between these different countries. In this article, the GDP of each country is forecasted using both past data of its own GDP, as well as the GDPs of the other European countries. The results show that the factor estimation models can be used as a helpful dimensionality reduction technique that generates predictions that are more accurate than those obtained using the benchmark model. This improvement is especially seen for predictions applying recursive estimation. It appears that the mean model is best in predicting the GDP for a rolling window forecasting method, whereas the boosting and ridge models dominate for the recursive estimation. Finally, it is noteworthy that the predictions seem to improve with the length of the forecast horizons. This, however, is not that surprising since Kim and Swanson (2018) come to the same conclusion.

This paper now proceeds as follows. Section 2 contains a literature review. In section 3 more insight is given into the gathering and transformation of the data. The fourth section starts with the introduction of the general prediction model and then touches upon the different methods that were used to construct and compare all forecasts. Then, section 5 presents and explains the findings. Finally, in section 6 we come to a conclusion on the results and discuss the limitations of our research.

2 Literature

The importance of big data in research is made clear through its extensive use in contemporary literature (Galicia et al., 2019; Hassani and Silva, 2015; Jeon et al., 2018; Shastri et al., 2019; Singh and Yassine, 2018; Torres et al., 2018). As is also addressed in the paper by Kim and Swanson (2018), it can be seen that many studies try and predict certain variables through the use of big data in factor models (Bok et al., 2018; Forni et al., 2000; Schiavoni et al., 2021; Waqar et al., 2017). The purpose of these factor models is to reduce the dimensionality of the data that is used. The dimensionality refers to the number of attributes in a data set, which for some data sets can easily become very large. Reducing the number of dimensions in the data often involves the reduction of the number of input variables or columns in the modelling data. The simplified data then produces a simpler regression model with fewer variables and a possibly better performance when making predictions on new data. Factor model analysis is therefore logically often applied to large high-dimensional data sets.

For the predictions of high frequency financial time series, it has already been proved that factor models are successful (Ait-Sahalia and Xiu, 2017; Back and Weigend, 1997; Oja et al., 2000). What is remarkable to note, however, is that Kim and Swanson (2018) analyse whether or not big data can also be put into use for the prediction of variables with a lower frequency. To do so, they investigate U.S. time series with monthly observations, including unemployment rates, inflation, and gross domestic product (GDP). In their research, the number of observations and variables are of the same order of magnitude. Usually, the suggested number of observations, when conducting factor analysis, equals between 3 and 20 times the number of variables (Mundfrom et al., 2005). This thus poses an interesting challenge.

A paper by Camba-Mendez et al. (2001) also applies a dynamic factor model to obtain forecasts for the output growth of four large European countries. They thus too investigate macroeconomic time series and apply factor analysis to deal with an excessive number of explanatory variables. However, in this paper the GDP growth is forecasted rather than the output growth and the analysis is applied to a larger set of European countries, also including smaller countries. Moreover, this paper uses only data on GDP to obtain the predictions. Finally, we consider two ways of extracting the underlying latent factors and combine these with several machine learning, variable selection and shrinkage methods.

3 Data

The data set that is used in this paper was obtained from AMECO (2021), which uses the database of Eurostat (2021) as its main source. As was explained earlier in the introduction section, this paper will investigate the predictability of the gross domestic product (GDP) of European countries. From this data set a total of 34 countries is included in the research, as displayed in table 1. If we take a look at the entire data set, it is noted that the earliest observations date from 1960 and the most recent observations date from 2020. However, when analysing the data for the individual countries specifically, it is found

that the availability of the data differs per country. The sample periods and the corresponding number of observations are also given in table 1. To account for the change in available data over time and to construct predictions for the countries with less observations, the methodology is adjusted accordingly. Section 4.1 elaborates on this.

Table 1

List of countries included in the research from the AMECO (2021) data set.

Country	**Code**	**Sample**	**Obsv.**	**Country**	**Code**	**Sample**	**Obsv.**
Albania	ALB	1996 - 2020	25	Lithuania	LTU	1993 - 2020	28
Austria	AUT	1960 - 2020	61	Luxembourg	LUX	1960 - 2020	61
Belgium	BEL	1960 - 2020	61	Malta	MLT	1991 - 2020	30
Bulgaria	BGR	1991 - 2020	30	Montenegro	MNE	2000 - 2020	21
Croatia	HRV	1995 - 2020	26	Netherlands	NLD	1960 - 2020	61
Cyprus	CYP	1990 - 2020	31	North Macedonia	MKD	1995 - 2020	26
Czechia	CZE	1990 - 2020	31	Norway	NOR	1960 - 2020	61
Denmark	DNK	1960 - 2020	61	Portugal	PRT	1960 - 2020	61
Estonia	EST	1993 - 2020	28	Romania	ROM	1990 - 2020	31
Finland	FIN	1960 - 2020	61	Serbia	SRB	2000 - 2020	21
France	FRA	1960 - 2020	61	Slovakia	SVK	1993 - 2020	28
Germany	DEU	1991 - 2020	30	Slovenia	SVN	1990 - 2020	31
Greece	GRC	1960 - 2020	61	Spain	ESP	1960 - 2020	61
Hungary	HUN	1991 - 2020	30	Sweden	SWE	1960 - 2020	61
Iceland	ISL	1960 - 2020	61	Switzerland	CHE	1960 - 2020	61
Ireland	IRL	1960 - 2020	61	Turkey	TUR	1960 - 2020	61
Italy	ITA	1960 - 2020	61	United Kingdom	GBR	1960 - 2020	61
Latvia	LVA	1990 - 2020	31				

In the paper by Kim and Swanson (2018) they find that it is important for time series to display stationarity. To ensure this for the time series used in this article, the following transformation is applied:

$$Y_t = \ln \frac{Z_{t+1}}{Z_t} \times 100.$$

Here, Z_t corresponds to the observation of the GDP at time t. The transformed time series Y_t is used in the analysis. This transformation does not only ensure stationarity, but also allows for a fair comparison of the GDP of the different countries. On top of the transformation that is applied, the article by Kim and Swanson (2018) mentions that the assumption is made that the time series are standardised, as is customary in its type of literature. Thus, in this paper, we make the same assumption about the standardisation of the transformed time series.

4 Methodology

In this research, a similar methodology is used as the one as explained in the work of Kim and Swanson (2018). We start by explaining the general model that is used for the data as well as the procedure to obtain the forecasts. These forecasts are obtained by means of a two-step procedure. Initially, the data is estimated using numerous dimensionality reduction techniques. These include principal component analysis (PCA) and independent component analysis (ICA), but also boosting, ridge regression, least angle regression, elastic net and non-negative garotte. These methods are explained in subsections 4.2 and 4.3. Secondly, several types of regressions are performed to obtain the forecasts and a set of benchmark models is created. Finally, the forecasts are compared using the Mean Square Forecast Error (MSFE).

4.1 General

In the research conducted by Kim and Swanson (2018) the following model is considered for the observed data:

$$X_{tj} = \Lambda_j' F_t + e_{tj}, \tag{1}$$

where X_{tj} denotes the observation of the GDP for the j-th country at time t. In this paper, there are a total of 34 investigated countries, such that $j = 1, ..., 34$. The number of observations t differs per country, as was explained earlier in the data section. Moving on, we have that F_t is a vector with r common factors, where $r < 34$. Then, Λ_j is the corresponding vector of factor loadings and e_{tj} is a vector with uncorrelated zero-mean disturbances. This paper investigates two methods of finding these underlying latent variables, as will be elaborated on in section 4.2.

Using this model, Kim and Swanson (2018) apply a two-step approach to obtain forecasts for the target variables. First, the data X_{tj} is used to obtain estimates for the factors F_t. Note that in this paper the countries all have different numbers of observations. These differences in availability of data thus cause for a different number of variables j at each point in time. Take for example time period $t = 30$, which corresponds to the year 1990. For 11 out of 34 countries there is no data available at this time at all and for another 5 countries we have just one observations at $t = 30$. These countries' data is thus not used to estimate the factors. The approach that is chosen to deal with the missing data values is the following. First, it is checked whether there is any observation at all for the target variable at time t. As will be explained later on in section 4.6, the so-called in-sample estimation period is chosen to have a minimum fixed length of 20. This means that predictions are formed using the past 20 observations, leading from $t - 1$ to $t - 20$. If there are no observations at all, no prediction is made for period $t + h$ and it is moved on to the next period. However, as soon as the threshold of 5 available observations is reached, the remaining $20 - 5 = 15$ observations are 'filled' with the mean of those 5 observations. In the next iteration, there are 14 observations missing, which are now filled with the mean of the 6 available values. This approach is implemented to ensure that predictions can be made at an earlier stage, even with a small amount of data available. Note that this measure is taken for the target variable Y_t. The other variables X_{tj} are included in the prediction of Y_t if they display no missing values at all in the sub sample. The variables X_{tj} that meet this condition, together with this newly formed Y_t, are then used to estimate the factors. Subsequently, these estimates $\hat{F}_t$ are used to obtain the estimators $\hat{\beta}_F$ and $\hat{\beta}_W$ using the following formula:

$$Y_{t+h} = W_t \beta_W + F_t \beta_F + \varepsilon_{t+h}. \tag{2}$$

Here, Y_t is the target variable to be predicted, which is the natural logarithm of the growth in GDP for country j at time t. The forecast horizon h is chosen to be 1, 3, or 5 years such that three different h-step ahead forecasts are made. The variable W_t is a vector of "additional" explanatory variables and F_t is a vector with r factors, where $r < 34$ for dimensionality reduction. Then, by regressing Y_{t+h} on $\hat{F}_t$ and W_t, the estimates for the coefficients, $\hat{\beta}_F$ and $\hat{\beta}_W$, are found.

In this paper, there are two methods applied to obtain the set of factors that is included in the model. These dimensionality reduction techniques are principal component analysis (PCA) and independent component analysis (ICA). Kim and Swanson (2018) evaluate several machine learning, variable selection and shrinkage methods on top of this. The methods that are also included in this research are boosting, ridge regression, least angle regression, the elastic net and non-negative garotte. Finally, Kim and Swanson (2018) also analyse combinations of these methods with the earlier dimensionality reduction techniques. All models are then used to construct forecasts for the target variables. To do so,

both a recursive as well as a rolling data window method are applied. The accuracy of these predictions is then evaluated using the MSFE. These methods and measure will also be used in this paper to build the models, create forecasts and compare the different countries with each other.

4.2 Diffusion Index Models

To predict the target variables and reduce the dimensionality of the data set, a dynamic factor model is put into work. The factors are estimated using two types of analyses: principal component analysis and independent component analysis. These estimated factors can be interpreted using diffusion indices. A diffusion index is a cross-sectional method to analyse common movement in a set of multiple time series (Vickrey, 2008). In sections 4.2.1 and 4.2.2 that follow it will be explained how these diffusion indices, or factors, are constructed.

4.2.1 Principal Component Analysis

To obtain a set of factors that best explains the variance in the model, PCA computes so-called *principal components* (PCs), which are linear combinations of the original variables as is shown in equation 1. The goal of finding these principal components is to extract the most important important information of the data set. These components are uncorrelated and sorted by the portion of variance they explain (Abdi and Williams, 2010). Here, the first principal component explains the largest portion of variance. This way the first few principal components account for a substantial amount of the variance explained and thus contain most of the relevant information. Then, by only using these first factors one reduces the dimensionality of the data. Although PCA is a very common approach to estimating latent factors, Kim and Swanson (2018) also elaborate on the use of ICA and SPCA instead. They explain some of the advantages and disadvantages that come along with each of the methods.

4.2.2 Independent Component Analysis

Another method that is used to reduce the dimensionality of the data is independent component analysis (ICA). In this case, the assumption is made that all time series depend on several unobserved and statistically independent diffusion indices. To find these independent components, we apply the FastICA algorithm in MATLAB from Gävert et al. (2021). The algorithm first centers the data by subtracting the mean. Then, the principal components are determined using PCA, which are then used to whiten the data. The whitening of data is the process of creating a new set of uncorrelated components that each have unit variance (Hossain, 2014). This centering and whitening of the data is called preprocessing. In general, ICA tries to find a unit weight vector w_j such that the projection matrix $w_j'X$ maximises nongaussianity. Nongaussianity - a measure of independence - is in turn measured using an approximation of its optimal estimator: negentropy (Hyvärinen, 2004). To approximate negentropy as a function of the estimated factors, denoted by $N(\hat{F})$, the formula by (Oja et al., 2000) is used:

$$N(\hat{F}) \approx \big(E[G(\hat{F})] - E[G(\nu)]\big)^2. \tag{3}$$

Here, ν is a standardised Gaussian variable and $G(.)$ is a non-quadratic function. To approximate negentropy correctly, it is important that the appropriate function for G is chosen. Oja et al. (2000) provide two suggestions, of which the following is used: $G(\hat{F}) = \frac{1}{a_1}\ln\cosh(a_1\hat{F})$ with $a_1 = 1$. The approximation of negentropy thus becomes:

$$N(\hat{F}) \approx \big(E[\ln\cosh(\hat{F})] - E[\ln\cosh(\nu)]\big)^2. \tag{4}$$

The FastICA algorithm by Gävert et al. (2021) enables you to specify this non-quadratic function G. It then uses this to find the independent components. An important difference between PCA and ICA is that the principal components obtained from PCA are ordered based on the variance that is explained by them. ICA, on the other hand, returns statistically independent factors for which it is not able to determine the explained portion of variance.

4.2.3 Number of Factors

In the research conducted by Kim and Swanson (2018) the number of factors that are included in the models is selected by means of a selection criterion from Bai and Ng (2002). However, when applying this criterion to the principal components of the data used in this paper, only the first factor was included in the analysis. This was true for each of the countries and at each point in time. Due to the outcome of this criterion, it was decided to use a different approach. In their paper, Kim and Swanson (2018) state that there are several other methods that can be considered for selecting the number of factors. This article therefore makes use of another more straightforward approach using the cumulative percentages of the total variance explained by the principal components. To do so, the change in cumulative percentage is calculated and the following formula is used:

$$\frac{\sum_{i=1}^{r} p_i - \sum_{i=1}^{r-1} p_i}{\sum_{i=1}^{r-1} p_i}. \tag{5}$$

Here, $\sum_{i=1}^{r} p_i$ denotes the cumulative percentage of the variance explained for the first r factors with $r = 1, ..., 34$. Evidently, the first factor is always included in the research. Then, the change in cumulative percentage is calculated for the consecutive factors. If this change is larger than a certain threshold, the factor is included in the models. The last factor r, for which the change in cumulative percentage is larger than this threshold, is included in the analysis. In this paper, the threshold is chosen to equal 0.05. The addition of factor $r+1$ thus does not improve the explained variance with more than 5% such factor r is the final factor that is included in the model. Note that the number of factors is reselected at each point in time and that the selection is done through the examination of the principal components. Then, ICA is applied using the same number of factors.

4.3 Machine Learning, Variable Selection and Shrinkage Methods

4.3.1 Boosting

'Boosting' is a machine learning based algorithm that is used to improve the accuracy of forecasts by decreasing a model's bias. The exact algorithm utilised in this paper is explained and elaborated on by Kim and Swanson in the working version of their paper (Kim and Swanson, 2013). What the boosting algorithm does, is that in each iteration it calculates the T-dimensional vector of 'current' residuals u. To initialise the algorithm, $\hat{\mu}_0$ is taken to equal $\bar{Z}$, where $Z_t = Y_t - \hat{Y}_t^W$. Here, $\hat{Y}_t^W$ is obtained by regressing Y_t on its lag Y_{t-1} and regressors W_t. In each iteration, the boosting algorithm calculates the elements of the vector of current residuals as: $u_t = Z_t - \hat{\mu}^{i-1}$. This vector is then regressed on each of the estimated factors $\hat{F}_j$ (for $j = 1, ..., r$) to obtain the coefficients $\hat{\beta}_j$. For each of the r factors the sum of squared residuals $SSR_j = (u - \hat{F}_j\hat{\beta}_j)'(u - \hat{F}_j\hat{\beta}_j)$ are compared. The factor that corresponds to the lowest sum of squared residuals, denoted by j^*, is then used to update the estimator as follows: $\hat{\mu}_i = \hat{\mu}_{i-1} + \nu\hat{F}_{j^*}\hat{\beta}_{j^*}$, where ν is the step length chosen to equal 0.800 in this paper. Note that this value is selected through a process of trial-and-error. To prevent over-fitting of the model, there should be a limit to the number of iterations performed by the boosting algorithm. Therefore, a stopping criterion is added, as described by Kim and Swanson (2013). The number of iterations M of the algorithm is selected such that:

$$M = \arg\min_{i} \left[\ln \sum_{t=1}^{T} \left(Y_t - \hat{\mu}_i\right)^2 + \frac{\ln T \times df^i}{T} \right], \tag{6}$$

where $df^i = trace(B^i)$ with $B^i = B^{i-1}\nu P^{(i)}(I_T - B^{i-1})$ and $P^{(i)} = \hat{F}_{j^*}\left(\hat{F}'_{j^*}\hat{F}_{j^*}\right)^{-1}\hat{F}_{j^*}$. Here, B^0 is given by a $(T \times 1)$-dimensional vector of ones.

After having run M iterations, the final estimator $\hat{\mu}^{Boost}$ is given by the boosting algorithm and the prediction of the target variable becomes:

$$Y_{t+h}^{Boost} = W_t\beta_W + \hat{\mu}^{Boost}. \tag{7}$$

4.3.2 Ridge Regression

One of the shrinkage methods that is discussed in this paper is the ridge regression. This is a penalised regression that is used to estimate the coefficients in multiple linear regression models. The method is used specifically to address any collinearity problems that might arise when explanatory variables are correlated to each other. The ridge regression estimates the coefficients in the model in equation 1 by minimising a penalised residual sum of squares. The following penalty function is used: $\eta \sum_{i=1}^{N} \beta_i^2$, where η is the penalty term. Solving this problem yields the following formula for the ridge coefficients:

$$\hat{\beta}_F^{Ridge} = (\hat{F}'\hat{F} + \eta\, I_N)^{-1}\hat{F}'Y. \tag{8}$$

Here, $\hat{F}$ denotes the matrix of estimated factors. Also, a penalty term of $\eta = 2$ is used in this paper, which is again chosen through trial-and-error. Note that an increasing η shrinks the coefficient estimates, but none are actually set to zero. To apply the ridge regression to the data, the ridge function from the Statistics and Machine Learning Toolbox from MATLAB is used (MATLAB, 2021). The final estimator $\hat{\beta}^{Ridge}$ is then used to calculate the prediction of the target variable:

$$Y_{t+h}^{Ridge} = W_t\beta_W + \hat{F}_t\hat{\beta}_F^{Ridge}. \tag{9}$$

4.3.3 Least Angle Regression

The least angle regression (LARS) is a variable selection method that finds the least absolute shrinkage and selection operator (lasso) (Tibshirani, 1996). Similar to the ridge regression, the lasso minimises a penalised residual sum of squares. The penalty function that is used by the lasso, however, is described as follows: $\eta \sum_{i=1}^{N} |\beta_i|$. The LARS algorithm that is used to find the lasso, applies a so-called 'forward-selection' approach and is described by Kim and Swanson (2013) in the working version of their paper. In principal, the algorithm creates a set of predictors by adding one explanatory variable at a time. In each step i, the previous estimator $\hat{\mu}^{i-1}$ is regressed onto the predictors that have not yet been added. In this paper, the total set of predictor variables consists of the estimated factors $\hat{F}_j$ (for $j = 1, ..., r$) obtained through the use of either PCA or ICA. Then, the adjusted coefficients of determination (adjusted R^2) of these regressions are compared and the factor j^* corresponding to the regression with the highest adjusted R^2 is added to the current active set. The estimator is then updated using this new set of explanatory variables. For the exact way of updating the estimator it is referred to the work of Kim and Swanson (2013). This process is continued until all variables have been added to the set of predictors.

The final LARS estimator is denoted by $\hat{\mu}^{LARS}$ and the prediction of the target variable is constructed as follows:

$$Y_{t+h}^{LARS} = W_t\beta_W + \hat{\mu}^{LARS} + \bar{Y}_t, \tag{10}$$

where $\bar{Y}_t$ is the mean of Y_t. Note that in the LARS method the target variable is standardised before the algorithm is applied. Therefore, the mean has to be added back to obtain the final prediction. It is noted that, although the variables were standardised beforehand (see section 3), a sub-sample has been taken to construct out-of-sample forecasts. Thus, it is still important to standardise the variables when applying LARS for the algorithm to work properly.

4.3.4 Elastic Net

Another method is the 'elastic net', which is both a variable selection method as well as a shrinkage method. The elastic net also minimises a penalised residual sum of squares and it uses two penalty functions to do so. These penalty functions are the ones used in the lasso and ridge regression methods. To apply the elastic net, the lasso function from the Statistics and Machine Learning Toolbox from MATLAB is used (MATLAB, 2021). In their documentation of the lasso function, the penalty function of the elastic net is described as: $\sum_{i=1}^{N}\left(\frac{1-\alpha}{2}\beta_i^2 + \alpha|\beta_i|\right)$. Given that a combination of the penalty functions of the lasso and ridge regression is applied, it is noted that the elastic net equals the lasso for $\alpha = 1$. Similarly, for α close to 0, the elastic net approaches the ridge regression.

Built into the lasso function from MATLAB (2021) is a possible K-fold cross-validation. In this paper, this builtin option is used and it is chosen to apply tenfold cross-validation, since this was also done by Zou and Hastie (2005). Finally, the estimated elastic net coefficients $\hat{\beta}_F^{EN}$ are used to construct Y_{t+h}^{EN} in the same way as was done for the ridge coefficients in equation 9.

4.3.5 Non-Negative Garotte

The non-negative garotte (NNG) estimator by Breiman (1995) is both a shrinkage as well as a variable selection method. Similar to the ridge regression, NNG shrinks coefficients of certain variables. However, the coefficients obtained from the ridge regression can only become very close to zero, whereas NNG has the ability to produce strictly zero coefficients. These variables are thus not taken into account in the regression, making NNG also a variable selection method.

The goal of NNG is to find the shrinkage factors $q(\zeta)$ by solving:

$$q(\zeta) = \underset{q(\zeta)}{\arg\min}\left\{\frac{1}{2T}\sum_{t=1}^{T}\left(Y_i - \sum_{j=1}^{N} G_j q_j\right)^2 + \zeta\sum_{j=1}^{N} q_j\right\} \quad \text{s.t. } q_j \geq 0 \tag{11}$$

for a given $\zeta > 0$ (Gijbels and Vrinssen, 2015). To do so, the algorithm of Kim and Swanson (2013) is applied. Similar to LARS, the non-negative garotte algorithm creates a current 'active' subset of regressors from the entire set of variables. In each iteration, the estimator is then updated until the final NNG estimator $\hat{\mu}^{NNG}$ is found. To construct the corresponding predictions, the same method is used as was done for the LARS estimator (equation 10).

4.3.6 Arithmetic Mean

Finally, a prediction model is formed by taking the arithmetic mean of the predictions obtained from the other models. The mean model is defined differently for each of the three specification types. These will be explained in section 4.5. The mean prediction always takes the mean of all benchmark models and the models that can be specified for that specification in particular. An overview of all models and their specifications is given in table 2.

4.4 Benchmark Models

In addition to the machine learning, variable selection and shrinkage methods, this paper also investigates a total of four benchmark models. These benchmark models are defined using either the data itself or the estimated factors from PCA or ICA.

4.4.1 Univariate Autoregression

The model that is used as the main benchmark in this paper is a univariate autoregressive model with p lags of the target variable as explanatory variables. To select the number of lags included in the AR(p) model, the Schwarz Information Criterion (SIC) is used. This criterion is specified as: $SIC(p) = \ln s_p^2 + p\frac{\ln N}{N}$, where p denotes the number of lags that are included in the model and s_p^2 the maximum likelihood estimator of the error variance in the model with p regressors. The SIC is a measure of fit of the model, containing a penalty term for the number of variables. The model with the smallest value of the SIC corresponds to the model with the best fit. After having specified the number of lags included in the model, the predictions for the target variable are constructed using least squares and the following formula:

$$Y_{t+h}^{AR} = \hat{\alpha} + \hat{\beta}_1 Y_{t-1} + ... + \hat{\beta}_p Y_{t-p}. \tag{12}$$

4.4.2 Multivariate Autoregression

The second benchmark model is the multivariate autoregressive model, ARX(p), which uses both lags of the target variable (Y_t) as well as lags of the other variables (X_{tj}) in the data set. To select the number of lags, we use the following procedure. First, the number of lags of the target variable is chosen to equal p as selected using the SIC for the univariate autoregressive model. Then, for $j = 1, ..., r$ the first lag $X_{t-1,j}$ is added to the model and the corresponding adjusted R^2 is calculated. If the addition of the variable increases the adjusted R^2 with more than 0.01, the variable $X_{t-1,j}$ is included in the model. Next, the second lag $X_{t-2,j}$ for each of the j variables individually. Again, the variable is retained if the adjusted R^2 increases with more than 0.01. This procedure is repeated for the first six lags of the variables, as was also done by Kim and Swanson (2018). Finally, the model is again estimated using least squares and the forecasts are computed as:

$$Y_{t+h}^{ARX} = \hat{\alpha} + \hat{\beta}_1 Y_{t-1} + ... + \hat{\beta}_p Y_{t-p} + \hat{\gamma} Z_t, \tag{13}$$

where Z_t contains the lags of X_{tj} that should included according to the previously explained procedure.

4.4.3 Principal Component Regression

The principal component regression (PCR) uses the estimated forecasts $\hat{F}_t$ obtained from PCA or ICA to calculate the prediction for the target variable. The forecasts are then formed through:

$$Y_{t+h}^{PCR} = \hat{\alpha} + \hat{\gamma} \hat{F}_t. \tag{14}$$

4.4.4 Factor Augmented Autoregression

The factor augmented autoregression (FAAR) applies a combination of the AR(p) model and the PCR by including lags of the target variable and the estimated factors as regressors. The predictions are given by:

$$Y_{t+h}^{FAAR} = \hat{\alpha} + \hat{\beta}_1 Y_{t-1} + ... + \hat{\beta}_p Y_{t-p} + \hat{\gamma}\hat{F}_t, \tag{15}$$

where the coefficients are estimated using least squares and the factors are estimated using either PCA or ICA.

4.5 Specification Types

As shown in equation 1, the target variable Y_{t+h} at time $t+h$ is predicted using data up to and including time t. In this paper, we specify three types of model specifications. These specifications differ in the data that is used to perform the regressions and make predictions for the target variable.

In their article Kim and Swanson (2018) discuss a total of six specification types for the construction of the forecasts. The first specification type is also investigated in this paper and is denoted by SP1. This specification type first estimates the underlying latent factors through PCA or ICA using the entire data set. Thus, the GDP data of all 34 countries are used to estimate the factors. Then, the number of factors that is included in the regressions is determined using the criterion as described earlier in section 4.2.3 of the methodology. These r factors are then used to create the prediction models. This article also looks into specification type SP1L for the machine learning methods in section 4.3. The difference between SP1 and SP1L is the addition of the lag of the estimated factors in the latter.

The third and final type of specification that is looked into in our research is denoted by SP3 and also obtained from the article by Kim and Swanson (2018). This specification type constructs predictions for the target variable without the estimation of the underlying latent factors. When taking a look at all of the methods and models described earlier, it is noted that there are three models that do not use the estimated factors and thus are specified according to SP3. These are the univariate autoregression, multivariate autoregression and arithmetic mean model.

Table 2

All models used to obtain predictions and the corresponding specification types for which they can be defined.

Abbrev.	**Methods**	**Specification**
AR	Univariate autoregression	SP3
ARX	Multivariate autoregression	SP3
PCR	Principal component regression	SP1
FAAR	Factor augmented autoregression	SP1
Boosting	Component boosting	SP1, SP1L
Ridge	Ridge regression	SP1, SP1L
LARS	Least angle regression	SP1, SP1L
EN	Elastic net	SP1, SP1L
NNG	Non-negative garotte	SP1, SP1L
Mean	Arithmetic mean	SP1, SP1L, SP3

4.6 Forecasting Methods

To construct these predictions, this paper applies two methods of forecasting: a rolling window and a recursive estimation method. These methods apply different approaches to choosing the observations that are used in the regression. A rolling window approach utilises a so-called 'rolling window' with fixed length. At each point in time the same number of past observations is used to construct a prediction for the target variable. In this paper, the length of this window is chosen to equal 20 observations. The recursive estimation method, on the contrary, uses all of the available past data to make predictions.

To compare the different forecasting methods with each other, it is chosen to lock the length of the prediction sample such that the number of predictions is equal for each of the different methods and forecast horizons. This length is chosen to equal 35 predictions. The in-sample estimation period is chosen to have a minimum fixed length of 20, also regardless of the length of the forecast horizon. This means that the first predictions can be made after $t = 20$. For the forecast horizon $h = 1$ a prediction is then made for time period $t = 21$, whereas an horizon of $h = 5$ means that a prediction is made for time $t = 25$.

4.7 Forecast Comparison

After having obtained the predictions from all of the different models (see table 2 for an overview), these should be compared in order to find the model with the best predictive performance. The measure that is used in this paper to compare the forecasts is the Mean Square Forecast Error (MSFE). The formula for the MSFE is given as follows (Kim and Swanson, 2018):

$$\text{MSFE} = \sum_{t=R-h+2}^{T-h+1} (Y_{t+h} - \hat{Y}_{t+h})^2, \tag{16}$$

where $\hat{Y}_{t+h}$ is the forecast for horizon h. From this definition of the MSFE it can easily be seen that a lower computed value of the MSFE corresponds to more accurate predictions.

5 Results

As is explained in the methodology section, the predictions are made for the target variable Y_t using a set of different models displayed in table 2. On top of these methods, there are three type of specification types considered in this paper: SP1, SP1L and SP3. Here, SP1 first estimates the factors and uses these to construct the predictions, whereas SP3 uses only data on GDP to make its predictions. Specification SP1L is similar to SP1, but also includes the first lag of the factors in the prediction models. Now, four of the benchmark models are defined for only one type of specification, being either SP3 or SP1. Then, the five machine learning methods are defined for both SP1 and SP1L. Finally, the arithmetic mean model can be specified for all three specifications. Given that there are 34 countries, three forecast horizons, two factor estimation methods and two methods of forecasting included in the research, this adds up to a total of $(4 \times 1 + 5 \times 2 + 1 \times 3) \times 3 \times 34 \times 2 \times 2 = 6936$ models being investigated in this paper. To easily compare the results, relative MSFEs are calculated. This is done by dividing a model's MSFE by the MSFE for the corresponding AR model. For example, take the predictions obtained from a ridge regression using components from PCA and a rolling window forecast with $h = 3$. Then, to calculate the relative MSFE, we divide its MSFE by the MSFE that corresponds to the AR model that also uses a rolling window forecast with $h = 3$. After calculating the relative MSFEs, these are used to determine what model generates the best predictions. This is the model with the lowest relative MSFE.

Tables 7 and 9 in the appendix display the results for the relative MSFEs using the rolling window and the recursive estimation methods respectively. The columns of the tables correspond to the different methods that are used to predict the target variable and the rows to the specific target variables. The table entries show the lowest relative MSFE that was obtained for the prediction models. The results are shown for each of the forecast horizons, specification types and factor estimation methods. One finds the method corresponding to the lowest relative MSFE in tables 8 and 10 for the rolling window and recursive estimation methods, respectively. For instance, if one takes a look at the first entry in the tables using a rolling window. This entry corresponds to the predictions of Albania's GDP using SP1, PCA and $h = 1$. In table 7 a value of 0.990 is found, which belongs to the MSFE of the predictions that were made using the arithmetic mean model (in table 8). Note that entries given in bold type denote the best relative MSFE per forecast horizon and country.

To get a clear overview of the performance of each of the different models for the different forecasting methods and specification types, the findings are summarised in tables 3 and 4. These differences will be discussed in subsections 5.1. On top of this, it is interesting to analyse whether there the predictive performance is higher for models using PCA than those using ICA. These dissimilarities are reviewed in section 5.2. Finally, it remains of interest to compare the individual countries to each other. To do so, table 6 summarises the results for the different countries and the different forecast horizons. These results are interpreted in section 5.3 and section 5.4, respectively.

5.1 Forecasting Methods

5.1.1 Rolling Window

To start off, we take a closer look at the results for the rolling window predictions (see appendix table 7). First of all, the portion of entries strictly lower than 1.000 can be calculated. If we do this, it is found that approximately 71.4% of the entries are below unity. Thus, for 71.4% of the models countries, the models generate better predictions than the univariate AR model that was used as a benchmark. Then, when observing the results for a forecast horizon of $h = 1$ in particular, it can be seen from table 3 below that most countries have a best MSFEs using specification SP1L. We start by looking at the values that are not within brackets. These values denote the number of times a model performs better than the other prediction models for that specific forecast horizon and specification. When focusing on the first specification type (SP1), it is seen that the results are quite consistent across the three forecast horizons. The mean model most often performs best, followed by the ridge and boosting models. The same is found for specification SP1L. Of all prediction models it thus appears to be the arithmetic mean model that produces the most favourable predictions.

It is also worth noting that of the machine learning, variable selection and shrinkage methods the LARS, EN and NNG models all perform very poorly. This holds for both specification types SP1 and SP1L. A possible explanation for the poor performance of the LARS and EN methods is that both are specifically effective in cases where the number of predictors is much larger than the number of observations (Efron et al., 2004; Zou and Hastie, 2005). In this paper, 34 countries with a maximum of 61 observations are considered, thus not meeting these requirements. About the non-negative garotte estimator it is stated by Kim and Swanson (2013) that: "The disadvantage of the NNG is its dependence on the ordinary least squares estimator, which can be especially problematic in small samples." If one takes a look at the size of the sub samples used in this paper, it is found that the total number of observations used to make predictions is indeed quite small. Kim and Swanson (2018) use a data set with a total

of 593 monthly observations, whereas this paper has a maximum of 61 annual observations. Also, our rolling window contains 20 observations, as opposed to the 156 observations in the work of Kim and Swanson. This thus seems like a plausible explanation for the worse performance of the NNG estimator. These findings regarding LARS, EN and NNG partially contradict those of Kim and Swanson (2018), who conclude that the methods combining factor estimation and machine learning methods dominate the ones that do not. In this research, however, this only appears to be true for boosting and ridge.

For the final specification type, it can be seen that the predictions using the AR model perform better than those using the mean model. Again, the results are consistent across the forecast horizons. It is notable that the ARX model never returns the lowest relative MSFE. However, it is not the case that this model never outperforms the AR model. This does happen sporadically, but apparently the mean model generates even better results in these cases. On top of this, it must be noted that the run time of the ARX model is much longer than that of the AR model. Obviously, the ARX model adds up to six lags of each variable and investigates whether it improves the adjusted R^2 enough. It is now found, however, that this time consuming process only improves the predictions sporadically and is thus not very beneficial.

Moving on, the values between brackets as given in table 3 count the number of times a model gives the best predictions amongst all specification types. By looking at these values, one can investigate which of the three specifications can best be used to obtain the forecasts. Specifications SP1 and SP1L seem to perform best with a total of 53 and 51 times, respectively. This confirms what Kim and Swanson (2018) say about the use of factors in prediction models, being that factor estimation methods can very well be used to obtain more accurate predictions. In their paper, Kim and Swanson conclude that the lags added in specification type SP1L are not useful for $h = 1$. When examining our results, it is also found that both the boosting model as well as the ridge regression decrease in performance when using SP1L compared to SP1. This thus supports the result of Kim and Swanson (2018). However, these differences are fairly small and this does not hold for the mean model, which actually increases in performance.

Table 3

Number of times each method gives the lowest relative MSFE for a country using a rolling window forecasting method. This is determined for each specification type and forecast horizon.

Specification	**SP1**				**SP1L**			
Horizon	**h = 1**	**h = 3**	**h = 5**	**all**	**h = 1**	**h = 3**	**h = 5**	**all**
FAAR	10 (1)	7 (1)	6 (2)	23 (4)	-	-	-	-
PCR	1 (0)	1 (1)	0 (0)	2 (1)	-	-	-	-
Boosting	14 (6)	12 (3)	18 (5)	44 (14)	8 (2)	9 (4)	13 (1)	17 (7)
Ridge	12 (5)	19 (4)	22 (8)	53 (17)	12 (2)	19 (7)	23 (6)	31 (15)
LARS	0 (0)	3 (0)	1 (0)	4 (0)	2 (0)	4 (0)	5 (0)	6 (0)
EN	2 (0)	1 (0)	0 (0)	3 (0)	3 (1)	0 (0)	0 (0)	3 (1)
NNG	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	1 (1)	0 (1)
Mean	31 (0)	27 (9)	23 (8)	81 (17)	45 (12)	38 (11)	28 (6)	83 (29)

Specification	**SP3**			
Horizon	**h = 1**	**h = 3**	**h = 5**	**all**
AR	18 (7)	19 (2)	27 (4)	64 (13)
ARX	0 (0)	0 (0)	0 (0)	0 (0)
Mean	17 (5)	16 (2)	8 (1)	41 (8)

Note. The values in brackets correspond to the number of times each method gives the best relative MSFE for a country, but now determined for each forecast horizon. A count between brackets thus correspond to a method giving the best prediction amongst all specification types (bold entries in appendix table 8), whereas a count outside brackets corresponds to a method giving the best prediction for that specification in particular (regular entries in appendix table 8).

5.1.2 Recursive Estimation

Now, if we compare these findings for the rolling window forecasting method with those for the recursive estimation, it is noted that the latter produces more prediction models with entries below unity. In table 9 about 82.4% of all entries are strictly lower than 1.000. It is thus noted that the recursive estimation method outperform the rolling window forecasting method. A possible reason for this could again be the size of the sub samples used in the rolling window method, as was discussed in the previous section. The rolling window contains 20 observations and might thus be too small for the models to perform properly. Since the recursive estimation method uses all available past observations, this perhaps is what causes the more desired results for the recursive estimation method.

The first thing that catches the eye when examining table 4 and comparing it to table 3, is that for specification SP1 the arithmetic mean model no longer performs best most often. This now clearly is the boosting model, again followed by the ridge regression. As opposed to the results for the rolling window method, this means that for the recursive estimation method a combination of factor estimation and machine learning methods does provide better predictions than those of a simple time series model. This finding is in line with the one of Kim and Swanson (2018). It could very well be the case that the increased number of observations with the recursive estimation method is again the reason behind this. It appears that the factor estimation and machine learning methods benefit more from these extra observations than the AR model does.

Similar to the results for the rolling data window method, there still are several machine learning, variable selection and shrinkage methods that perform poorly. However, now it is noted that for specification SP1L the least angle regression and the elastic net show an increased performance. As opposed to the findings for the rolling window method, it is now seen that the LARS and EN models give the best predictions 11 and 18 times, respectively. This increased performance may again be explained by the increased number of observations with the recursive estimation. The models appear to improve as the number of observations increases. Finally, we again observe the values within brackets to compare the different specification types to each other. These show that SP1 clearly outperforms the other specifications. Comparing this to the findings of Kim and Swanson (2018), it is confirmed once more that the factor estimation methods can be used to obtain more accurate predictions. Also, the inclusion of lags as is done in specification SP1L does not produce favorable results.

Table 4

Number of times each method gives the lowest relative MSFE for a country using a recursive estimation forecasting method. These are determined for each specification type and forecast horizon.

Specific.	**SP1**				**SP1L**			
Horizon	**h = 1**	**h = 3**	**h = 5**	**all**	**h = 1**	**h = 3**	**h = 5**	**all**
FAAR	15 (5)	7 (2)	7 (4)	29 (11)	-	-	-	-
PCR	2 (1)	1 (0)	1 (0)	4 (1)	-	-	-	-
Boosting	18 (11)	28 (16)	29 (15)	75 (42)	10 (0)	13 (1)	15 (3)	23 (4)
Ridge	15 (7)	16 (5)	23 (7)	54 (19)	15 (2)	25 (6)	38 (6)	40 (14)
LARS	1 (0)	2 (0)	1 (0)	4 (0)	6 (0)	5 (0)	2 (0)	11 (0)
EN	4 (0)	0 (0)	0 (0)	4 (0)	11 (1)	7 (2)	1 (0)	18 (3)
NNG	0 (0)	0 (0)	2 (0)	2 (0)	0 (0)	0 (0)	0 (0)	0 (0)
Mean	15 (4)	16 (0)	7 (0)	38 (4)	28 (2)	20 (2)	14 (0)	48 (4)

Specific.	**SP3**			
Horizon	**h = 1**	**h = 3**	**h = 5**	**all**
AR	9 (2)	12 (0)	19 (0)	40 (2)
ARX	0 (0)	0 (0)	0 (0)	0 (0)
Mean	26 (3)	23 (2)	16 (0)	65 (5)

Note. The values in brackets correspond to the number of times each method gives the best relative MSFE for a country, but now determined for each forecast horizon. A count between brackets thus correspond to a method giving the best prediction amongst all specification types (bold entries in appendix table 10), whereas a count outside brackets corresponds to a method giving the best prediction for that specification in particular (regular entries in appendix table 10).

5.2 Factor Estimation Methods

In section 4.2 the diffusion index models have been explained. To estimate the factors in such models, there are two methods used: principal component analysis (PCA) and independent component analysis (ICA). The theoretical differences between PCA and ICA have been made clear, but the empirical differences are yet to be investigated. To do so, the number of times is counted that either PCA or ICA is the preferred method of factor estimation by checking which of the two generates lower MSFEs. This is done for the two specification types (SP1 and SP1L) separately and the counts are given in table 5.

In their paper, Kim and Swanson (2018) find that ICA and SPCA provide more accurate predictions for a forecast horizon of $h = 1$. However, for the longer forecast horizons PCA is the preferred factor estimation method. This is due to the nature of PCA, being more robust to structural breaks at a shorter forecast horizon. Similar results are obtained in this paper. By looking at the results for specification SP1L, it is found that the performance of PCA increases with the length of the forecast horizon. Also, for the shortest forecast horizon $h = 1$ our results confirm that ICA is the preferred method, whereas the PCA is preferred for the longest forecast horizon. However, the results for specification type SP1 appear to be less consistent. If we zoom in on the rolling window method, the differences between PCA and ICA are fairly small. If anything, ICA is the preferred method. This is also true for the longer forecast horizons. But, due to the larger number of observations, it may again be more valuable to focus on the recursive estimation method. Here, PCA is preferred to ICA.

Table 5

Number of times PCA performs better than ICA (or vice versa) by producing a lower best relative MSFE. This is determined for each forecasting method, specification type and forecast horizon.

	Specific.	**SP1**				**SP1L**			
	Horizon	**h = 1**	**h = 3**	**h = 5**	**all**	**h = 1**	**h = 3**	**h = 5**	**all**
Rolling	PCA	16	17	15	48	11	14	19	44
	ICA	18	17	19	54	23	20	15	58
Recursive	PCA	21	29	22	72	14	23	25	62
	ICA	13	5	12	30	20	11	9	40

5.3 Countries

Having examined the differences between the rolling data window method and the recursive estimation method, it is also interesting to gain insight into possible differences between countries. Therefore, table 6 displays the averages of the relative MSFEs for each of the countries and forecast horizons. This enables us to quickly detect for which of the countries the models perform better.

When investigating the averages for all forecast horizons combined, there seems to be quite some variation between the countries. The average relative MSFE for Latvia is only 0.536, whereas Austria has an average of 1.139. We start by looking at this low value for Latvia, which can be caused by one of the following two things. Either the univariate AR model for Latvia performs poorly, leading to exceptionally high relative MSFEs for the other prediction models, or the other prediction models simply perform very well. This could for example be caused by the characteristics of the data for Latvia. Perhaps the principal components do a really good job explaining data due to high correlations between the PCs and the data. The opposite reasoning can be applied to the higher value of the average MSFE for Austria: either the AR model produces accurate predictions or the other models perform poorly. Moving on, it is also noted that for 27 out of 34 countries the average is below 1.000, meaning that for most countries the prediction models provide more accurate forecasts than the AR model. For the other countries, however, it is more beneficial to just obtain the forecasts using a simple AR model.

If these results are set against the ones by Kim and Swanson (2018), it is especially striking that their findings show less values above unity. Again, this may be due to the limited number of observations used to make the predictions in this paper. Using a larger data set allows the machine learning methods to extract more information from the data and with this improve the accuracy of the forecasts.

Moving on, it is of interest to gain insight into the effect of the availability of the data on the results. As was explained earlier in the data section, there are some countries for which the available data does not reach from 1960 until 2020 (see table 1). In section 4.1 it is explained how these missing values are approached so that forecasts can still be generated for these countries. The question now remains whether or not this influences the accuracy of the predictions. The countries with missing data observations are denoted with an asterisk in table 6. This creates two groups for which the total average can be calculated from the averages per country (in column 'all'). Starting with the rolling window method, these averages are 0.721 and 0.963 for the countries with and without asterisks, respectively. For the recursive estimation, these averages are 0.732 and 0.897. The relative MSFEs are thus lower in the group of countries with an asterisk, suggesting that the predictions are more accurate for the countries with less available data. The most obvious reason would again be poor performance of the AR model for these countries.

Table 6

The averages of the best relative MSFEs for both forecasting methods, given for each country and forecasting horizon.

	Rolling				**Recursive**			
Horizon	**h = 1**	**h = 3**	**h = 5**	**all**	**h = 1**	**h = 3**	**h = 5**	**all**
ALB*	0.997	0.832	0.747	0.859	0.823	0.839	0.667	0.776
AUT	1.224	1.006	1.187	1.139	1.013	0.939	1.026	0.993
BEL	1.222	0.980	0.979	1.060	1.008	0.897	0.925	0.943
BGR*	0.803	0.687	0.512	0.667	0.760	0.700	0.489	0.650
HRV*	0.750	0.862	0.750	0.787	0.806	0.791	0.811	0.803
CYP*	0.793	0.680	0.744	0.739	0.819	0.769	0.711	0.766
CZE*	0.658	0.705	0.796	0.720	0.725	0.771	0.792	0.762
DNK	0.967	1.053	0.986	1.002	1.101	1.026	1.044	1.057
EST*	0.667	0.804	0.784	0.751	0.642	0.690	0.757	0.697
FIN	0.865	0.896	0.922	0.894	0.763	0.895	0.781	0.813
FRA	1.103	0.995	1.009	1.036	0.997	0.906	0.985	0.963
DEU*	0.700	0.618	0.717	0.678	0.780	0.708	0.824	0.771
GRC	0.852	0.921	0.906	0.893	0.943	0.821	0.809	0.858
HUN*	0.980	0.667	0.653	0.767	0.940	0.794	0.765	0.833
ISL	1.074	0.974	1.065	1.038	0.721	0.736	0.680	0.712
IRL	0.824	0.881	0.748	0.818	0.878	0.842	0.771	0.831
ITA	1.071	1.063	1.004	1.046	0.978	0.937	0.898	0.938
LVA*	0.599	0.564	0.445	0.536	0.586	0.584	0.452	0.541
LTU*	0.863	0.527	0.463	0.617	0.731	0.526	0.453	0.570
LUX	0.882	0.861	0.962	0.902	0.857	0.810	0.908	0.859
MLT*	0.732	0.778	0.967	0.826	0.804	0.916	1.005	0.908
MNE*	0.542	0.541	0.702	0.595	0.581	0.496	0.688	0.588
NLD	0.980	0.982	1.137	1.033	0.886	0.919	1.004	0.936
MKD*	0.744	0.562	0.404	0.570	0.775	0.640	0.440	0.619
NOR	0.926	0.778	0.766	0.823	0.883	0.911	0.942	0.912
PRT	1.221	0.852	0.857	0.977	1.066	0.856	0.728	0.883
ROM*	0.707	0.712	0.742	0.720	0.769	0.762	0.784	0.772
SRB*	0.936	0.728	0.581	0.749	0.763	0.876	0.601	0.747
SVK*	0.845	0.842	0.822	0.836	0.751	0.784	0.742	0.759
SVN	0.778	0.898	0.857	0.844	0.794	1.008	0.837	0.880
ESP	0.978	0.977	0.972	0.976	1.001	0.862	0.897	0.920
SWE	1.087	0.918	0.950	0.985	0.782	1.010	0.779	0.857
CHE	0.813	1.003	0.889	0.902	0.780	0.853	0.807	0.813
TUR	0.738	0.782	0.719	0.746	0.746	0.886	0.769	0.801
GBR	1.090	1.045	1.034	1.056	1.202	0.981	1.008	1.064
Average	0.886	0.828	0.822	0.845	0.842	0.821	0.788	0.817

Note. The countries with asterisks denote the countries in the data set for which not all observations were available. These countries thus have a lower number of total observations than those without an asterisk.

5.4 Forecast Horizons

The last row of table 6 displays the average relative MSFE for each of the three forecast horizons. It is noted that the averages corresponding to the rolling window method are higher than those of the recursive estimation method. With the definition of the MSFE in mind, this thus suggests that recursive estimation can be used to make better predictions. Again note that this may be caused by the limited data that is used in this paper. In their paper, Kim and Swanson (2018) conclude that the recursive estimation

method is best for the shorter forecast horizons, whereas the rolling data window method appears to produce the most favorable outcomes for the longer forecast horizons. In this paper it has already been established that the recursive estimation dominates the rolling window method for all results. The rolling window method is thus not the preferred method, not even for the longer forecast horizons.

When investigating the averages for the different forecast horizons, the following is found. For both the rolling window as well as the recursive estimation method, the average MSFE corresponding to the forecasts using $h = 1$ is higher than that of the longer forecast horizons. This is surprising, since one would expect it to be more difficult to make predictions further in the future rather than just one time period ahead. However, Kim and Swanson (2018) also note in their paper that the machine learning and shrinkage methods perform better for the longer forecast horizons. Possible structural breaks in the data seem to have a relatively bigger negative effect on the predictions obtained through the simple AR model than it has on the predictions using the machine learning methods. This thus provides an explanation for the smaller relative MSFEs for the longer forecast horizons.

6 Conclusion

This paper partially replicates the work of Kim and Swanson (2018). Several dimensionality reduction techniques have been put into use to investigate the predictability of the growth in GDP of 34 European countries. Both principal component analysis and independent component analysis were applied as well as a number of machine learning, variable selection and shrinkage methods. These methods included boosting, ridge regression, least angle regression, elastic net and non-negative garotte. These models were compared to a set of benchmark models: univariate autoregression, multivariate autoregression, principal component regression and factor augmented autoregression. In total there are 6936 models investigated. To evaluate the results of these models, the Mean Square Forecast Error is used. In addition to the replication of the work that is done by Kim and Swanson (2018), a solution is found for applying the methodology to a data set with missing observations.

When analysing the results, it is found that the factor estimation methods in general dominate the univariate autoregressive model that is used as a benchmark. This supports the findings of Kim and Swanson (2018) on the usefulness of factor estimation. Respectively 71.4% and 82.4% of the predictions for the rolling data window method and the recursive estimation method are more accurate than the predictions obtained using the AR model. The difference between these two models may be explained by the length that was used for the rolling window. The window was chosen to be too small and thus contains too little information. An increase in the number of observations, as is done with the recursive estimation, generates more accurate predictions.

Evaluation of the results for the different prediction models and specification types allows us to find the model that is most successful in predicting the growth in GDP. For the rolling data window method it appears to be the mean model that performs best. This partially supports the usefulness of factor estimation, but contradicts the statement of Kim and Swanson (2018) on the benefits of model averaging methods. Fortunately, more favorable results are found for the recursive estimation method. Here, the boosting algorithm and ridge regression turn out to be the best in predicting the GDP growth. These methods make use of factor estimation, but are not model averaging models. Therefore, these results are in line with the work of Kim and Swanson (2018).

Although these results are in agreement with the work of Kim and Swanson, it is notable that some of the machine learning, variable selection and shrinkage methods outperform the benchmark AR model only occasionally. These methods are LARS, EN and NNG. Possibly due to the number of predictor variables being smaller than the number of observations in our data set, the LARS and EN method are not effective. Also, the NNG model depends on the ordinary least squares estimator, causing problems in small samples, thus providing a plausible explanation for the lack of performance of the NNG estimator. Despite these unfavorable results regarding the performance of some models, it noteworthy that both LARS and EN clearly show increased performance for SP1L and recursive estimation. Although not further analysed, it could be that an even larger number of observations improves the results even more.

When comparing the results for the countries with each other, the differences are somewhat difficult to interpret. For some of the countries, the methods do a really good job in predicting the GDP, whereas for others one can better stick to a simple AR model. In the end, there are 27 out of 34 countries for which the predictions are improved when compared to the AR model. As for the differences in results for the three forecast horizons, it is noted that the predictions improve with the length of the forecast horizon. Although this seems counter-intuitive, it actually is in line with the work of Kim and Swanson (2018). It appears that possible structural breaks in the data set have a bigger negative effect on the predictions obtained using a simple AR model than it has on those obtained using machine learning methods. This explains why the relative MSFEs are smaller for longer forecast horizons.

Overall, it can be concluded that the investigated models can often be used to successfully predict the GDP of European countries. The results are thus consistent with the findings of Kim and Swanson (2018). Nonetheless, this paper does know some limitations. The first limitation involves the selection of the number of factors. The method that was proposed by Kim and Swanson (2018) resulted in the inclusion of only the first factor in the analysis. Therefore, an other simple approach was applied leading to a higher number of factors being selected. This method, however, has not been investigated extensively in literature. On top of this, the penalty terms in both the boosting algorithm as well as the ridge regression are selected by means of trial-and-error. Although the boosting and ridge models do end up to perform well, this approach is not an efficient nor sophisticated one. It could have been useful to gain more insight into the selection of these penalty parameters and set criteria with which these parameters are chosen. Finally, the biggest limitation in this research lies with the selection and processing of the data set. A solution for the low number of observations that was used to make the predictions, could be to select a larger in-sample estimation period. However, this would probably only increase the performance slightly. It is even more beneficial to utilise a larger data set with more observations. For instance, a data set that contains monthly data on the GDP of European countries over the same period from 1960 until 2020.

References

H. Abdi and L. J. Williams. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459, 2010.

Y. Ait-Sahalia and D. Xiu. Using principal component analysis to estimate a high dimensional factor model with high-frequency data. *Journal of Econometrics*, 201(2):384–399, 2017.

AMECO, Apr 2021. URL `https://ec.europa.eu/info/business-economy-euro/indicators-statistics/economic-databases/macro-economic-database-ameco/ameco-database_en`. Accessed: 10-05-2021.

A. D. Back and A. S. Weigend. A first application of independent component analysis to extracting structure from stock returns. *International journal of neural systems*, 8(04):473–484, 1997.

J. Bai and S. Ng. Determining the number of factors in approximate factor models. *Econometrica*, 70 (1):191–221, 2002.

B. Bok, D. Caratelli, D. Giannone, A. M. Sbordone, and A. Tambalotti. Macroeconomic nowcasting and forecasting with big data. *Annual Review of Economics*, 10:615–643, 2018.

L. Breiman. Better subset regression using the nonnegative garrote. *Technometrics*, 37(4):373–384, 1995.

G. Camba-Mendez, G. Kapetanios, R. J. Smith, and M. R. Weale. An automatic leading indicator of economic activity: forecasting GDP growth for European countries. *The Econometrics Journal*, 4(1): S56–S90, 2001.

A. De Mauro, M. Greco, and M. Grimaldi. A formal definition of Big Data based on its essential features. *Library Review*, 2016.

F. X. Diebold and R. S. Mariano. Comparing predictive accuracy. *Journal of Business & economic statistics*, 20(1):134–144, 2002.

B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *The Annals of statistics*, 32 (2):407–499, 2004.

Eurostat, May 2021. URL `https://ec.europa.eu/eurostat/databrowser/bookmark/6c5e015e-0448-431a-b94c-f6c9aa43d071?lang=en`. Accessed: 10-05-2021.

M. Forni, M. Hallin, M. Lippi, and L. Reichlin. The generalized dynamic-factor model: Identification and estimation. *Review of Economics and statistics*, 82(4):540–554, 2000.

A. Galicia, R. Talavera-Llames, A. Troncoso, I. Koprinska, and F. Martínez-Álvarez. Multi-step forecasting for big data time series based on ensemble learning. *Knowledge-Based Systems*, 163:830–841, 2019.

I. Gijbels and I. Vrinssen. Robust nonnegative garrote variable selection in linear regression. *Computational Statistics & Data Analysis*, 85:1–22, 2015.

H. Gävert, J. Hurri, J. Särelä, and A. Hyvärinen. Fastica, 2021. URL `https://research.ics.aalto.fi/ica/fastica/`. Accessed: 20-05-2021.

H. Hassani and E. S. Silva. Forecasting with big data: A review. *Annals of Data Science*, 2(1):5–19, 2015.

M. Hossain. Whitening and coloring transforms for multivariate Gaussian random variables, 2014. URL https://www.projectrhea.org/rhea/index.php/ECE662_Whitening_and_Coloring_Transforms_S14_MH. Accessed: 20-06-2021.

A. Hyvärinen. Negentropy, Apr 2004. URL http://cis.legacy.ics.tkk.fi/aapo/papers/IJCNN99_tutorialweb/node14.html. Accessed: 19-06-2021.

S. Jeon, B. Hong, and V. Chang. Pattern graph tracking-based stock price prediction using big data. *Future Generation Computer Systems*, 80:171–187, 2018.

H. H. Kim and N. R. Swanson. Mining big data using parsimonious factor and shrinkage methods. *Available at SSRN 2294110*, 2013.

H. H. Kim and N. R. Swanson. Mining big data using parsimonious factor, machine learning, variable selection and shrinkage methods. *International Journal of Forecasting*, 34(2):339–354, 2018.

M. MATLAB. Statistics and Machine Learning Toolbox R2021a, 2021. URL https://nl.mathworks.com/products/statistics.html#resources. Accessed: 13-06-2021.

D. J. Mundfrom, D. G. Shaw, and T. L. Ke. Minimum sample size recommendations for conducting factor analyses. *International Journal of Testing*, 5(2):159–168, 2005.

E. Oja, K. Kiviluoto, and S. Malaroiu. Independent component analysis for financial time series. In *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No. 00EX373)*, pages 111–116. IEEE, 2000.

C. Schiavoni, F. Palm, S. Smeekes, and J. van den Brakel. A dynamic factor model approach to incorporate big data in state space models for official statistics. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 184(1):324–353, 2021.

M. Shastri, S. Roy, and M. Mittal. Stock price prediction using artificial neural model: an application of big data. *EAI Endorsed Transactions on Scalable Information Systems*, 6(20), 2019.

S. Singh and A. Yassine. Big data mining of energy time series for behavioral analytics and energy consumption forecasting. *Energies*, 11(2):452, 2018.

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

J. F. Torres, A. Galicia, A. Troncoso, and F. Martínez-Álvarez. A scalable approach based on deep learning for big data time series forecasting. *Integrated Computer-Aided Engineering*, 25(4):335–348, 2018.

A. Vickrey. Moody's analytics buffet blog, 2008. URL https://www.economy.com/support/blog/buffet.aspx?did=7FEA3BF2-D05D-41B7-A288-C88D0E413445. Accessed: 15-06-2021.

M. Waqar, H. Dawood, P. Guo, M. B. Shahnawaz, and M. A. Ghazanfar. Prediction of stock market by principal component analysis. In *2017 13th International Conference on Computational Intelligence and Security (CIS)*, pages 599–602. IEEE, 2017.

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

Appendices

Table 7

The best relative MSFEs using a rolling window forecasting method. These are given for each country, specification type and forecast horizon, individually.

Horizon	**h = 1**					**h = 3**					**h = 5**				
Method	**SP1**		**SP1L**		**SP3**	**SP1**		**SP1L**		**SP3**	**SP1**		**SP1L**		**SP3**
	PCA	**ICA**	**PCA**	**ICA**		**PCA**	**ICA**	**PCA**	**ICA**		**PCA**	**ICA**	**PCA**	**ICA**	
ALB	0.990	0.990	1.030	1.030	**0.949**	0.782	0.831	**0.766**	0.833	0.948	0.664	0.682	**0.656**	0.731	1.000
AUT	1.232	1.251	1.320	1.320	**1.000**	1.004	1.004	1.012	1.012	**0.996**	1.235	1.235	1.232	1.232	**1.000**
BEL	1.212	1.212	1.343	1.343	**1.000**	1.014	**0.907**	1.011	1.011	0.958	0.988	0.983	**0.965**	**0.965**	0.993
BGR	**0.746**	0.765	0.757	0.760	0.988	**0.542**	0.649	0.544	0.720	0.981	0.323	0.405	**0.319**	0.514	1.000
HRV	0.687	**0.634**	0.777	0.777	0.874	0.784	0.867	**0.781**	0.879	1.000	0.719	**0.631**	0.712	0.686	1.000
CYP	0.796	0.796	**0.743**	**0.743**	0.888	0.612	**0.525**	0.597	0.667	1.000	**0.639**	0.704	0.645	0.731	1.000
CZE	0.569	0.699	0.574	**0.562**	0.887	**0.573**	0.723	0.595	0.694	0.940	**0.679**	0.709	0.708	0.884	1.000
DNK	0.935	**0.896**	1.003	1.003	1.000	1.037	1.059	1.084	1.084	**1.000**	**0.965**	0.987	0.989	0.989	1.000
EST	0.609	0.630	**0.596**	0.632	0.866	0.728	0.828	**0.711**	0.753	1.000	**0.637**	0.713	0.659	0.910	1.000
FIN	0.858	0.858	**0.844**	**0.844**	0.918	**0.827**	**0.827**	0.914	0.914	0.995	**0.894**	0.913	0.902	0.903	1.000
FRA	1.114	1.027	1.187	1.187	**1.000**	1.050	1.016	**0.955**	**0.955**	1.000	1.033	**0.952**	1.031	1.031	1.000
DEU	0.632	0.622	0.659	**0.587**	1.000	**0.472**	0.542	0.526	0.555	0.993	0.717	0.603	0.720	**0.543**	1.000
GRC	0.922	0.929	0.812	0.812	**0.788**	0.931	0.931	**0.873**	**0.873**	0.996	0.934	0.934	**0.845**	**0.845**	0.971
HUN	**0.911**	0.930	0.944	1.116	1.000	0.644	0.565	0.637	**0.533**	0.958	**0.526**	0.575	0.532	0.631	1.000
ISL	1.098	1.098	1.090	1.090	**0.995**	0.994	0.994	**0.941**	**0.941**	1.000	1.075	1.089	1.042	1.119	**1.000**
IRL	**0.698**	0.822	0.773	0.827	1.000	0.831	0.839	**0.829**	0.908	1.000	0.678	0.672	**0.659**	0.729	1.000
ITA	1.059	1.068	1.170	1.110	**0.950**	1.051	1.110	1.079	1.079	**0.993**	1.060	1.027	0.988	0.988	**0.956**
LVA	**0.426**	0.621	0.443	0.622	0.882	0.455	0.519	0.481	**0.367**	1.000	0.299	**0.272**	0.303	0.353	1.000
LTU	**0.832**	0.852	0.832	0.832	0.967	**0.398**	0.413	0.414	0.409	1.000	0.318	0.361	**0.311**	0.373	0.950
LUX	0.868	0.868	**0.838**	**0.838**	1.000	**0.789**	**0.789**	0.828	0.900	1.000	**0.925**	**0.925**	0.948	1.015	1.000
MLT	0.662	**0.614**	0.641	0.741	1.000	0.712	0.722	**0.708**	0.748	1.000	0.938	0.976	**0.928**	0.994	1.000
MNE	**0.378**	0.378	0.426	0.547	0.983	**0.416**	0.473	0.432	0.585	0.799	0.613	0.758	0.596	**0.542**	1.000
NLD	0.982	0.982	0.998	0.998	**0.940**	0.990	0.990	**0.964**	**0.964**	1.000	1.150	1.150	1.192	1.192	**1.000**
MKD	0.724	0.724	**0.685**	**0.685**	0.902	0.464	0.522	0.468	**0.430**	0.925	0.269	**0.232**	0.288	0.276	0.956
NOR	0.906	0.906	**0.893**	0.922	1.000	**0.657**	0.720	0.725	0.788	1.000	**0.701**	0.750	0.707	0.726	0.946
PRT	1.227	1.258	1.311	1.311	**1.000**	0.831	0.831	**0.823**	**0.823**	0.951	**0.771**	**0.771**	0.872	0.874	0.998
ROM	0.659	0.696	**0.618**	0.675	0.886	0.558	0.708	**0.555**	0.759	0.978	**0.624**	0.661	0.649	0.774	1.000
SRB	**0.909**	0.940	0.915	0.918	1.000	0.597	0.781	**0.548**	0.724	0.989	0.474	**0.430**	0.474	0.529	1.000
SVK	**0.775**	0.822	0.777	0.850	1.000	0.814	0.850	0.817	**0.750**	0.979	0.759	**0.757**	0.778	0.817	1.000
SVN	**0.686**	0.751	0.704	0.748	1.000	0.898	**0.805**	0.894	0.894	1.000	0.779	0.830	**0.770**	0.906	1.000
ESP	0.991	1.000	1.009	1.009	**0.884**	1.006	0.965	**0.957**	**0.957**	1.000	0.986	0.986	**0.943**	**0.943**	1.000
SWE	1.116	1.116	1.113	1.090	**1.000**	**0.877**	**0.877**	0.918	0.918	1.000	**0.935**	**0.935**	0.939	0.939	1.000
CHE	0.836	0.836	**0.766**	**0.766**	0.861	**0.974**	**0.974**	1.033	1.033	1.000	**0.847**	**0.847**	0.876	0.876	1.000
TUR	0.688	**0.674**	0.688	0.680	0.962	**0.702**	**0.702**	0.773	0.783	0.949	**0.625**	0.683	0.636	0.651	1.000
GBR	1.140	1.140	1.085	1.085	**1.000**	1.097	1.097	1.015	1.015	**1.000**	1.047	1.047	1.039	1.039	**1.000**

Note. The rows correspond to the countries for which the GDP is predicted and the columns to the different methods used to predict these variables. The table entries show the lowest relative MSFE that was obtained for the prediction models. Table 8 show the methods for which the lowest MSFE is found. Entries given in bold type denote the best relative MSFE per forecast horizon and country.

Table 8

Methods for which the best relative MSFEs is found using a rolling window forecasting method. These are given for each country, specification type and forecast horizon, individually.

Horizon			**h = 1**					**h = 3**					**h = 5**		
Method	**SP1**		**SP1L**		**SP3**	**SP1**		**SP1L**		**SP3**	**SP1**		**SP1L**		**SP3**
	PCA	**ICA**	**PCA**	**ICA**		**PCA**	**ICA**	**PCA**	**ICA**		**PCA**	**ICA**	**PCA**	**ICA**	
ALB	Mean	Mean	Mean	Mean	**Mean**	Ridge	Mean	**Ridge**	Mean	Mean	Ridge	Boost	**Ridge**	Boost	AR
AUT	NNG	Mean	Mean	Mean	**AR**	Mean	Mean	Mean	Mean	**Mean**	Mean	Mean	Mean	Mean	**AR**
BEL	Mean	Mean	Mean	Mean	**AR**	Mean	**FAAR**	Mean	Mean	Mean	Mean	FAAR	**Mean**	**Mean**	Mean
BGR	**Boost**	Mean	Ridge	Mean	Mean	**Ridge**	Boost	Ridge	Boost	Mean	Ridge	Boost	**Ridge**	Boost	AR
HRV	Boost	**Ridge**	Mean	Mean	AR	Ridge	EN	**Ridge**	EN	AR	Ridge	**Ridge**	Ridge	Ridge	AR
CYP	Mean	Mean	**Mean**	**Mean**	Mean	Ridge	**Ridge**	Ridge	Boost	AR	**Ridge**	Boost	Ridge	EN	AR
CZE	Boost	Boost	Ridge	**Boost**	Mean	**Boost**	Ridge	Ridge	Mean	Mean	**Ridge**	Boost	Ridge	Mean	AR
DNK	FAAR	**FAAR**	Mean	Mean	AR	FAAR	FAAR	Mean	Mean	**AR**	**FAAR**	FAAR	Mean	Mean	AR
EST	Ridge	PCR	**Ridge**	Mean	Mean	Ridge	EN	**Ridge**	Ridge	AR	**Boost**	Boost	Boost	Boost	AR
FIN	Mean	Mean	**Mean**	**Mean**	Mean	**Mean**	**Mean**	Mean	Mean	Mean	**Ridge**	Mean	Ridge	Mean	AR
FRA	FAAR	FAAR	Mean	Mean	**AR**	FAAR	FAAR	**Mean**	**Mean**	AR	FAAR	**FAAR**	Mean	Mean	Mean
DEU	Ridge	Boost	Ridge	**Boost**	AR	**Boost**	Boost	Ridge	Boost	Mean	Ridge	Boost	Ridge	**Boost**	AR
GRC	FAAR	Mean	Mean	Mean	**AR**	Mean	Mean	**Mean**	**Mean**	Mean	Mean	Mean	**Mean**	**Mean**	Mean
HUN	**Ridge**	Boost	Ridge	Boost	AR	Ridge	Boost	Ridge	**Boost**	Mean	**Boost**	Boost	Ridge	Boost	AR
ISL	Mean	Mean	Mean	Mean	**Mean**	Mean	Mean	**Mean**	**Mean**	AR	Ridge	Mean	Ridge	Mean	**AR**
IRL	**Boost**	Boost	Boost	Mean	AR	Ridge	Ridge	**Ridge**	Mean	AR	Ridge	Boost	**LARS**	Boost	AR
ITA	NNG	FAAR	Mean	NNG	**Mean**	FAAR	FAAR	Mean	Mean	**Mean**	Mean	FAAR	Mean	Mean	**Mean**
LVA	**Boost**	Ridge	Boost	Mean	Mean	Ridge	Boost	Ridge	**Boost**	AR	Ridge	**Boost**	Ridge	Boost	AR
LTU	**FAAR**	Mean	Mean	Mean	Mean	**Ridge**	Boost	Ridge	Boost	AR	Ridge	Boost	**Ridge**	Boost	Mean
LUX	Mean	Mean	**Mean**	**Mean**	AR	**Mean**	**Mean**	EN	Mean	AR	**Mean**	**Mean**	EN	EN	AR
MLT	Ridge	**Ridge**	Ridge	EN	AR	Ridge	EN	**Ridge**	EN	AR	Ridge	EN	**Ridge**	EN	AR
MNE	**Boost**	Boost	Ridge	EN	Mean	**PCR**	Ridge	Ridge	Mean	AR	Ridge	Ridge	Ridge	**Ridge**	AR
NLD	Mean	Mean	Mean	Mean	**Mean**	Mean	Mean	**Mean**	**Mean**	AR	Mean	Mean	Mean	Mean	**AR**
MKD	Mean	Mean	**Mean**	**Mean**	Mean	Ridge	Boost	Ridge	**Boost**	Mean	Ridge	**Boost**	Ridge	Boost	Mean
NOR	Mean	Mean	**NNG**	Mean	AR	**Boost**	Boost	EN	Mean	AR	**Boost**	Mean	Ridge	Mean	Mean
PRT	FAAR	Mean	Mean	Mean	**AR**	Mean	Mean	**Mean**	**Mean**	Mean	**Mean**	**Mean**	EN	Mean	Mean
ROM	Ridge	Ridge	**Ridge**	Ridge	Mean	Ridge	Ridge	**Ridge**	Mean	Mean	**Ridge**	Boost	Ridge	Mean	AR
SRB	**Ridge**	Mean	Ridge	Mean	AR	Boost	Mean	**Boost**	Boost	Mean	Ridge	**Ridge**	Ridge	Boost	AR
SVK	**Ridge**	Boost	Ridge	Boost	AR	Ridge	Boost	Ridge	**Ridge**	Mean	Boost	**Ridge**	Ridge	Ridge	AR
SVN	**Boost**	Ridge	Ridge	Boost	AR	Mean	**Ridge**	Mean	Mean	AR	Ridge	Boost	**Ridge**	Boost	AR
ESP	FAAR	FAAR	Mean	Mean	**Mean**	Mean	NNG	**Mean**	**Mean**	AR	Mean	Mean	**Mean**	**Mean**	AR
SWE	Mean	Mean	Mean	NNG	**AR**	**Mean**	**Mean**	Mean	Mean	AR	**Mean**	**Mean**	Mean	Mean	AR
CHE	Mean	Mean	**Mean**	**Mean**	Mean	**Mean**	**Mean**	Mean	Mean	AR	**Mean**	**Mean**	Mean	Mean	AR
TUR	Mean	**Boost**	Mean	Boost	Mean	**Mean**	**Mean**	Ridge	Mean	Mean	**Ridge**	Boost	Ridge	Boost	AR
GBR	Mean	Mean	Mean	Mean	**AR**	Mean	Mean	Mean	Mean	**AR**	Mean	Mean	Mean	Mean	**AR**

Note. The rows correspond to the countries for which the GDP is predicted and the columns to the different methods used to predict these variables. The table entries show the model for which the lowest relative MSFE was obtained for the prediction models. Table 7 show these relative MSFEs. Entries given in bold type denote the method corresponding to the best relative MSFE per forecast horizon and country.

Table 9

The best relative MSFEs using a recursive estimation forecasting method. These are given for each country, specification type and forecast horizon, individually.

Horizon			**h = 1**					**h = 3**					**h = 5**		
Method	**SP1**		**SP1L**		**SP3**	**SP1**		**SP1L**		**SP3**	**SP1**		**SP1L**		**SP3**
	PCA	**ICA**	**PCA**	**ICA**		**PCA**	**ICA**	**PCA**	**ICA**		**PCA**	**ICA**	**PCA**	**ICA**	
ALB	0.861	**0.727**	0.808	0.808	0.911	0.769	0.791	**0.761**	0.935	0.937	0.598	0.591	0.596	**0.547**	1.000
AUT	0.992	1.004	1.117	1.083	**0.871**	**0.884**	0.953	0.972	0.996	0.890	0.932	**0.922**	1.152	1.177	0.944
BEL	**0.901**	0.944	1.138	1.138	0.920	**0.784**	0.964	0.884	0.999	0.854	**0.782**	0.994	0.894	1.045	0.908
BGR	0.712	0.712	**0.696**	**0.696**	0.987	**0.572**	0.660	0.584	0.719	0.963	0.301	0.419	**0.291**	0.433	1.000
HRV	0.762	0.755	0.824	**0.729**	0.962	0.747	0.747	**0.714**	0.749	1.000	0.715	0.797	**0.703**	0.838	1.000
CYP	**0.733**	0.815	0.824	0.824	0.897	**0.667**	0.708	0.668	0.804	1.000	0.648	**0.629**	0.643	0.635	1.000
CZE	**0.636**	0.685	0.678	0.693	0.932	0.730	**0.643**	0.714	0.771	0.995	0.733	**0.711**	0.743	0.773	1.000
DNK	**0.981**	1.116	1.144	1.264	1.000	**0.941**	1.063	1.085	1.041	1.000	**0.895**	0.963	1.153	1.211	1.000
EST	0.584	**0.569**	0.588	0.581	0.889	0.613	0.616	0.614	**0.608**	1.000	0.675	**0.650**	0.691	0.771	1.000
FIN	0.776	0.776	0.808	**0.610**	0.845	**0.834**	0.875	0.902	1.001	0.862	0.668	0.733	**0.664**	0.877	0.961
FRA	0.924	0.953	1.191	1.009	**0.907**	**0.837**	0.858	1.001	0.952	0.883	0.980	**0.896**	1.081	1.018	0.949
DEU	**0.659**	0.695	0.684	0.862	1.000	0.556	0.730	**0.550**	0.710	0.994	**0.737**	0.802	0.759	0.822	1.000
GRC	0.954	0.965	1.025	1.025	**0.746**	0.781	0.859	0.829	**0.765**	0.870	**0.651**	0.853	0.777	0.915	0.850
HUN	0.879	**0.878**	0.939	1.002	1.000	**0.663**	0.820	0.676	0.841	0.967	0.530	0.900	**0.527**	0.865	1.000
ISL	**0.686**	0.744	0.690	0.713	0.770	**0.684**	0.733	0.708	0.757	0.797	0.556	0.649	**0.521**	0.724	0.950
IRL	**0.820**	0.869	0.873	0.886	0.944	**0.776**	0.782	0.796	0.857	1.000	**0.638**	0.744	0.672	0.802	1.000
ITA	**0.850**	0.945	1.171	1.061	0.863	0.926	0.931	1.017	0.982	**0.831**	0.908	**0.769**	1.041	0.913	0.859
LVA	**0.452**	0.589	0.481	0.495	0.910	**0.408**	0.546	0.452	0.518	0.996	**0.272**	0.312	0.290	0.401	0.986
LTU	**0.671**	**0.671**	0.675	0.675	0.966	**0.388**	0.433	0.419	0.392	1.000	**0.320**	0.328	0.336	0.320	0.960
LUX	**0.824**	0.834	0.828	0.872	0.928	0.707	0.806	**0.699**	0.872	0.967	**0.801**	0.932	0.890	0.918	1.000
MLT	**0.695**	0.767	0.723	0.837	1.000	0.831	0.959	**0.789**	0.999	1.000	0.985	1.069	**0.964**	1.006	1.000
MNE	**0.400**	0.584	0.400	0.566	0.953	0.404	**0.365**	0.426	0.594	0.689	0.633	0.584	0.640	**0.583**	1.000
NLD	0.940	**0.792**	0.941	0.952	0.806	**0.857**	0.931	0.907	0.941	0.960	0.942	**0.931**	1.059	1.097	0.991
MKD	0.794	**0.648**	0.758	0.758	0.918	0.502	**0.495**	0.505	0.751	0.948	0.313	**0.284**	0.337	0.294	0.973
NOR	**0.790**	0.872	0.904	0.904	0.943	**0.774**	0.978	0.848	0.955	1.000	**0.886**	0.985	0.939	0.946	0.952
PRT	1.092	**0.888**	1.272	1.115	0.961	**0.794**	0.873	0.810	0.902	0.899	**0.569**	0.726	0.713	0.729	0.902
ROM	**0.720**	0.762	0.727	0.732	0.903	0.672	0.708	**0.671**	0.761	1.000	**0.676**	0.721	0.681	0.840	1.000
SRB	**0.595**	0.801	0.661	0.760	1.000	**0.800**	0.868	0.811	0.901	1.000	**0.461**	0.532	0.470	0.543	1.000
SVK	0.694	0.670	0.726	**0.666**	1.000	**0.711**	0.739	0.717	0.833	0.918	0.735	0.622	0.749	**0.616**	0.990
SVN	**0.684**	0.732	0.747	0.807	1.000	1.016	1.032	**0.997**	**0.997**	1.000	0.771	**0.760**	0.808	0.845	1.000
ESP	**0.864**	1.027	1.155	1.088	0.873	0.853	0.853	0.867	**0.741**	0.993	**0.800**	0.870	0.916	0.900	1.000
SWE	**0.749**	**0.749**	0.784	0.772	0.857	**0.936**	1.008	1.056	1.056	0.994	**0.666**	0.824	0.705	0.800	0.901
CHE	0.760	0.785	0.769	0.831	**0.753**	**0.770**	0.875	0.896	0.797	0.926	**0.645**	0.774	0.799	0.890	0.925
TUR	**0.655**	0.719	0.711	0.711	0.935	**0.844**	0.880	0.874	0.882	0.951	**0.624**	0.772	0.666	0.784	1.000
GBR	1.235	1.076	1.363	1.334	**1.000**	0.989	1.002	1.051	0.945	**0.920**	**0.902**	1.085	0.928	1.123	1.000

Note. The rows correspond to the countries for which the GDP is predicted and the columns to the different methods used to predict these variables. The table entries show the lowest relative MSFE that was obtained for the prediction models. Table 10 show the methods for which the lowest MSFE is found. Entries given in bold type denote the best relative MSFE per forecast horizon and country.

Table 10

Methods for which the best relative MSFEs is found using a recursive estimation forecasting method. These are given for each country, specification type and forecast horizon, individually.

Horizon	**h = 1**					**h = 3**					**h = 5**				
Method	**SP1**		**SP1L**		**SP3**	**SP1**		**SP1L**		**SP3**	**SP1**		**SP1L**		**SP3**
	PCA	**ICA**	**PCA**	**ICA**		**PCA**	**ICA**	**PCA**	**ICA**		**PCA**	**ICA**	**PCA**	**ICA**	
ALB	FAAR	**Boost**	Mean	Mean	Mean	Ridge	Boost	**Ridge**	Mean	Mean	Boost	Ridge	Boost	**Ridge**	AR
AUT	FAAR	NNG	Mean	NNG	**Mean**	**Boost**	Mean	Boost	Mean	Mean	Boost	**FAAR**	Ridge	Mean	Mean
BEL	**FAAR**	FAAR	Mean	Mean	Mean	**FAAR**	Mean	Boost	Mean	Mean	**Boost**	Mean	Boost	Mean	Mean
BGR	Mean	Mean	**Mean**	**Mean**	Mean	**Ridge**	Boost	Ridge	EN	Mean	Ridge	Ridge	**Ridge**	Ridge	AR
HRV	Mean	Ridge	Mean	**Ridge**	Mean	Ridge	Ridge	**Ridge**	Ridge	AR	Ridge	Boost	**Ridge**	Ridge	AR
CYP	**Boost**	Mean	Mean	Mean	Mean	**Ridge**	PCR	Ridge	Ridge	AR	Ridge	**Boost**	Ridge	Boost	AR
CZE	**Ridge**	Boost	Ridge	Boost	Mean	Ridge	**Boost**	Ridge	Ridge	Mean	Ridge	**Ridge**	Ridge	Ridge	AR
DNK	**Boost**	FAAR	Boost	Boost	AR	**Boost**	FAAR	Boost	NNG	AR	**Boost**	FAAR	Ridge	Mean	AR
EST	Boost	**Ridge**	Boost	Ridge	Mean	Boost	Ridge	Ridge	**Ridge**	AR	Ridge	**Boost**	Ridge	Ridge	AR
FIN	Mean	Mean	Mean	**NNG**	Mean	**Boost**	Mean	Boost	Mean	Mean	Boost	FAAR	**Boost**	Mean	Mean
FRA	FAAR	NNG	Mean	NNG	**Mean**	**FAAR**	FAAR	Mean	NNG	Mean	Boost	**FAAR**	Mean	Ridge	Mean
DEU	**Ridge**	PCR	Ridge	EN	AR	Ridge	EN	**Ridge**	EN	Mean	**Ridge**	Boost	Ridge	Boost	AR
GRC	FAAR	Mean	Mean	Mean	**AR**	Boost	Mean	Boost	**NNG**	Mean	**Boost**	Mean	Boost	Mean	Mean
HUN	Ridge	**PCR**	Ridge	EN	AR	**Ridge**	Mean	Ridge	Mean	Mean	Ridge	PCR	**Ridge**	Mean	AR
ISL	**Boost**	Ridge	Boost	NNG	Mean	**Boost**	Mean	Boost	Mean	Mean	Boost	Ridge	**Boost**	Boost	Mean
IRL	**Boost**	NNG	Ridge	NNG	Mean	**Boost**	Boost	Ridge	Boost	AR	Ridge	LARS	**Ridge**	Mean	AR
ITA	**FAAR**	NNG	NNG	NNG	Mean	FAAR	FAAR	Boost	NNG	**Mean**	FAAR	**FAAR**	Boost	NNG	Mean
LVA	**Boost**	Ridge	Boost	Ridge	Mean	**Boost**	Boost	Boost	Ridge	Mean	**Ridge**	Boost	Ridge	Boost	Mean
LTU	**Mean**	**Mean**	Mean	Mean	Mean	**Ridge**	Boost	Ridge	Boost	AR	**Boost**	Boost	Ridge	Ridge	Mean
LUX	**Boost**	Mean	EN	Mean	Mean	Boost	Boost	**Boost**	Mean	Mean	**Boost**	Mean	Ridge	Mean	AR
MLT	**Ridge**	Ridge	Ridge	Ridge	AR	Ridge	EN	**Ridge**	EN	AR	LARS	EN	**Ridge**	EN	AR
MNE	**Ridge**	EN	Ridge	EN	Mean	Ridge	**Boost**	Ridge	Ridge	AR	Ridge	Boost	Ridge	**Boost**	AR
NLD	FAAR	**FAAR**	EN	Mean	Mean	**Boost**	Mean	EN	Ridge	Mean	Boost	**FAAR**	Boost	Ridge	Mean
MKD	Ridge	**Ridge**	Mean	Mean	Mean	Ridge	**Boost**	Ridge	Mean	Mean	Ridge	**Ridge**	Ridge	Ridge	Mean
NOR	**Boost**	Boost	Mean	Mean	Mean	**Boost**	Mean	Ridge	Mean	AR	**Boost**	Mean	Ridge	Mean	Mean
PRT	FAAR	**FAAR**	Mean	NNG	Mean	**Boost**	Mean	EN	Mean	Mean	**Boost**	Ridge	EN	Ridge	Mean
ROM	**Ridge**	Mean	Ridge	Mean	Mean	Ridge	Boost	**Ridge**	Ridge	AR	**Ridge**	Boost	Ridge	Ridge	AR
SRB	**Boost**	Ridge	Boost	Ridge	AR	**Ridge**	Mean	Boost	Mean	AR	**Ridge**	Boost	Ridge	Ridge	AR
SVK	Boost	Ridge	Ridge	**Ridge**	AR	**Boost**	Boost	Ridge	Mean	Mean	Ridge	Ridge	Ridge	**Ridge**	Mean
SVN	**Boost**	Boost	Ridge	Boost	AR	Ridge	Mean	**Mean**	**Mean**	AR	Ridge	**Boost**	Ridge	Boost	AR
ESP	**FAAR**	FAAR	EN	NNG	Mean	Mean	Mean	**NNG**	NNG	Mean	**Boost**	Mean	Ridge	Ridge	AR
SWE	**Mean**	**Mean**	Mean	NNG	Mean	**Boost**	FAAR	Mean	Mean	Mean	**Boost**	Mean	Boost	Mean	Mean
CHE	Boost	Mean	Boost	Mean	**Mean**	**Boost**	Mean	Mean	Boost	Mean	**Boost**	Ridge	Ridge	Mean	Mean
TUR	**Boost**	Mean	Mean	Mean	Mean	**Boost**	Mean	Ridge	Mean	Mean	**Boost**	Boost	Ridge	Mean	AR
GBR	Boost	FAAR	Boost	NNG	**AR**	Boost	Mean	Mean	NNG	**Mean**	**Boost**	Mean	Boost	Mean	AR

Note. The rows correspond to the countries for which the GDP is predicted and the columns to the different methods used to predict these variables. The table entries show the model for which the lowest relative MSFE was obtained for the prediction models. Table 9 show these relative MSFEs. Entries given in bold type denote the method corresponding to the best relative MSFE per forecast horizon and country.