ÁDÁM CSÁPAI
University of Economics in Bratislava, Slovakia

MACROECONOMIC FORECASTING USING MACHINE LEARNING: A CASE OF SLOVAKIA

Abstract:

We assess the forecasting performance of the selected machine learning methods. According to previous research, they can enhance short-term forecasting performance. We forecast industrial production, inflation and unemployment in Slovakia. We compare the forecasting performance of the models using the mean absolute error and root-mean-squared error. We forecast the variables using ensemble machine learning techniques, such as random forest, bagging and boosting. Additionally, we explore regularized least squares models, such as ridge regression, lasso regression, and elastic net models. Finally, we examine the forecasting performance of neural networks and compare the mean and trimmed mean of model forecasts with individual model performance. Our findings affirm that these methods can enhance forecast accuracy of short-term forecasts, particularly when a relatively large dataset is available. Individual machine learning models prove themselves to be even more accurate than the averages of model forecasts.

Keywords:

Economic forecasting, Slovakia, Ensemble machine learning, Regularized least squares, Neural networks

JEL Classification: C53, E37, E27

1 Introduction

Accurate monitoring and forecasting of macroeconomic variables are crucial for informed economic decision-making. Methods of forecasting economic phenomena using big data have evolved over time. These techniques strive to emulate, elucidate, and automate the most effective forecasting practices employed by investment markets, central banks, and other market monitoring endeavors.

Forecasting plays a crucial role in evaluating the state and direction of the economy. Government budgets depend on forecasted macroeconomic variables, such as GDP, inflation and unemployment. Policymakers, such as the Central Bank use forecasting models to accurately time policy interventions.

This paper focuses on showcasing diverse forecasting methods based on big data and compares their performance to a benchmark linear regression model. Our primary aim is to see whether individual machine learning models can outperform the traditional econometric model. Our secondary aim is to compare the performance of individual forecasting models to their combined mean and trimmed mean values. The subsequent sections delve into the literature review, the presentation of the models, the results obtained, and a concluding summary, respectively.

2 Literature review

Friedman et al. (2001) and James et al. (2013) extensively discuss the utilization of predictive tools for processing big data. However, the use of big data poses certain challenges in the field of modeling, such as result distortion and the potential generation of false positive outcomes. Nevertheless, Baldacci et al. (2016) states that leveraging big data offers the advantage of providing timely information about the state of the economy without being subjected to subsequent revisions.

There are multiple papers which use machine learning methods for economic forecasting. Özgür and Akkoc (2021) forecast inflation rates using shrinkage methods of machine learning algorithms (ridge, lasso, ada lasso and elastic net) and compare them to multiple benchmarks. They find that lasso and elastic net algorithms outperform conventional econometric methods in the case of Turkish inflation.

Richardson et al. (2018) analyzes the real-time nowcasting performance of machine learning algorithms estimated on New Zealand data. They train these algorithms to nowcast quarterly GDP growth. They find that machine learning models outperform statistical models and that combining the individual machine learning nowcasts further improves performance.

Kurihara and Fukushima (2019) examine the validity of forecasting economic variables by using machine learning methods. They use a long-short-term memory neural network to forecast GDP and consumer price in the G7 countries. They find that the neural network is less appropriate for GDP forecasting, but there is no difference between the benchmark statistical models and the network.

Medeiros et al. (2019) attempt to forecast U.S. inflation using a novel dataset. They show that ML models with a large number of covariates are systematically more accurate than the benchmarks. They highlight the performance of the random forest model, which outperforms all other models. They conclude that the performance improvement is likely due to the potential nonlinearities between past key macroeconomic variables and inflation.

Based on this section we conclude that machine learning models have great potential for being a useful tool in macroeconomic forecasting. This potential has not yet been tapped for the CEE, and especially for the V4 economies. As a first step of a long-term research project, we apply a large set of machine learning methods to forecast Slovak industrial production, inflation and unemployment to see whether these methods perform as well in Slovakia as in other countries.

3 Methodology and methods

In this section we describe our forecasting models and accuracy measures. The detailed description of these methods is in Maehashi and Shintani (2020). To begin with, we write our benchmark linear regression model as

$$y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \varepsilon_i = \mathbf{x}_i^\top \boldsymbol{\beta} + \varepsilon_i,\ i = 1, \dots, n, \tag{1}$$

Since this model is pretty standard, we assume that the reader is familiar with the model properties. If not, please refer to Greene (2003), who describes the method in detail.

3.1 Regularized least squares

To continue with, we describe the regularized least squares methods, otherwise known as penalized regression methods, used for reducing the number of predictors. These methods minimize

$$\sum_{t=1}^{T} \left\{ \left(y_{t+h} - \sum_{i=1}^{N} \beta_i x_{it} \right)^2 + \lambda J(\beta) \right\}, \tag{2}$$

where the regularization hyperparameter λ plays a crucial role. Regularized least squares methods define the penalty term $J(\beta)$, where $\beta = (\beta_1, \beta_2, \dots, \beta_N)'$, in different ways. To determine the appropriate value of λ for all methods, we use 5-fold cross-validation as in Maehashi and Shintani (2020). In this paper we present and use the ridge, lasso and elastic net regressions.

Lasso. The first method is the lasso regression, otherwise known as the “least absolute shrinkage and selection operator” introduced by Tibshirani (1996). Lasso specifies the penalty function as $J(\beta) = \sum_{i=1}^{N} |\beta_i|$ by using the L_1 norm as the penalty term. This formulation introduces a kink at 0 in a constrained minimization problem concerning β. Consequently, numerous coefficient estimates become zero. Hence, lasso is regarded as a variable selection procedure.

Ridge. Hoerl and Kennard (1970) introduce the ridge regression, which uses an L_2 norm penalty function $J(\beta) = \sum_{i=1}^{N} \beta_i^2$. In the case of the ridge regression, the regression coefficients can approach zero without being exactly equal to it. This implies that ridge regression applies a shrinkage method to mitigate overfitting by shrinking the coefficients.

Elastic net. Zou and Hastie (2005) show that lasso outperforms ridge when a significant number of model coefficients tend to be zero, whereas ridge is more appropriate when predictors are highly correlated. To get the best of both worlds, we also introduce elastic net. It provides the advantages of both methods simultaneously by introducing a penalty function denoted as $J(\beta) = \omega \sum_{i=1}^{N} |\beta_1| + (1-\omega) \sum_{i=1}^{N} \beta_1^2$, where ω represents a hyperparameter that determines the relative weights of the L_1 norm penalty and the L_2 norm penalty. The value of ω is determined through cross-validation, which we do as in Maehashi and Shintani (2020). Notably, when ω is set to 1, elastic net is equivalent to lasso, while setting ω to 0 renders it equivalent to a ridge regression. Consequently, elastic net encompasses both shrinkage and selection characteristics.

3.2 Ensemble machine learning

In this paper we use decision trees, otherwise known as regression trees, to account for potential nonlinear relationships within the data. One of our primary objectives is to capture and incorporate these nonlinearities. The process of constructing these trees involves grouping similar observations by iteratively generating nodes within the tree.

In our forecasting framework, we consider a target variable denoted as y_{t+h}, which represents the specific macroeconomic time series we aim to predict. Initially, each observation of y_{t+h} is allocated to nodes based on a selected predictor variable $X_i = (x_1, x_{2x}, \dots, x_N)$. The nodes that are selected receive the value of the sample mean y_{t+h}, conditional on the specific predictor. If there are any remaining nodes that have not been assigned values, we further partition them using the remaining predictors. This process continues until we assign appropriate values to all of the nodes.

A regression tree with *M* terminal nodes can be written as

$$y_{t+h} = \sum_{m=1}^{M} \theta_m 1_{[x_i \in h_m]} + \varepsilon_{i+h_m}, \tag{3}$$

where we denote the indicator function as $1_{[x_i \in h_m]}$ and R_m is a region which is a subset of the space of X_b. Meanwhile, θ_m represents the sample mean of the target variable y_{t+h} conditional on $X_t \in R_m$. The objective of the estimation is to identify the tree structure that minimizes $\sum_{i=1}^{T} \varepsilon_{i+h}^2$. To accomplish this, we select sorting variables from X_t and determine splitting values at each node. This is done by an algorithm developed by Breiman et al. (1984) to identify the optimal sorting and splitting values.

Regression trees exhibit strong performance when there are nonlinearity and variable interactions in the data. However, their ability to accurately forecast is often suboptimal due to their sensitivity to changes in the dataset. In order to address this issue, we do not solely rely on regression trees themselves. Instead, we use them as the foundation for our ensemble machine learning techniques, namely bagging, boosting, and random forests.

By using ensemble methods, we are able to overcome the limitations of individual regression trees. Bagging, boosting, and random forests leverage multiple regression trees to improve predictive accuracy and enhance robustness. These methods combine the predictions of multiple trees to make more reliable and stable forecasts. This way, we mitigate the sensitivity of individual trees to variations in the data, resulting in improved out-of-sample forecast performance.

Bagging. Bagging, short for bootstrap aggregating, is a technique that has demonstrated its ability to enhance forecast accuracy through empirical exercises conducted by Breiman (1996). Bühlmann and Yu (2002) show that bagging effectively reduces forecast errors for independent and identically distributed (i.i.d.) data. Inoue and Kilian (2008) do the same exercise with time series data. In the bagging approach, we generate ***B*** bootstrap samples of the predictor variables $X_t = (x_i, x_2, \dots, x_N)'$ and the target variable y_{t+h}. For each bootstrap sample $X_t^{(b)}$ and $y_{t+h}^{(b)}$, we construct a regression tree to obtain a forecast $\hat{y}_{t+h}^{(b)}$. Finally, we aggregate the forecasts by averaging them as $B^{-1}\sum_{b=1}^{B}\hat{y}_{t+h}^{(b)}$. This averaging process helps alleviate the issues of overfitting and excessive volatility associated with individual forecasts. In our empirical forecasting exercise ***B*** = 10.

Boosting. Schapire (1990) proposes boosting as a method to address the issue of overfitting. Let us consider $\sum_{m=1}^{M}\theta_m 1_{[X_t \in R_m]}$, which represents a simple regression tree with an initial value of $f_0(X_i) = \eta\sum_m^M \theta_m 1_{[X_t \in R_m]}$. Here, $\eta \in (0,1)$ is a learning rate parameter, which we set at $\eta = 0.1$. In boosting, it is desirable to have shallow regression trees, meaning that each base learner $f_s(X_t)$, for $s = 0,1, \dots S$, should be a weak learner. During each stage, the boosting algorithm incorporates information from previous trees by considering the forecast errors. It searches for a new algorithm with an L_2 loss function based on this error information. The model is updated at the s -th stage according to

$$f_s(X_t) = f_{s-1}(X_t) + \eta \sum_{m=1}^{M_s} \theta_{sm} \mathbf{1}_{[X_t \in R_{sm}]}, \tag{4}$$

Here, $\sum_{m=1}^{M_s}\theta_{sm}\mathbf{1}_{[X_t \in R_{sm}]}$ is estimated for the residual obtained from the $(s-1)$-th stage, i.e., $y_{t+h} - f_{s-1}(X_t)$. This process continues until we reach the specified limit on the number of boosting stages.

Random forests. Breiman (2001) introduces random forests as a variation of bagging. While bagging forecasts remain stable as long as the regression trees generated from different bootstrap samples are not highly correlated, the situation changes if there is significant correlation

among the trees. In such cases, simply averaging the predictions may not effectively reduce forecast variance since the individual trees in the bootstrap samples tend to be similar.

To address this issue, Hastie et al. (2009) proposes a dropout procedure to decorrelate the regression trees in individual samples. Specifically, the set of predictors $X_i = (x_{11}, x_{21}, \dots, x_{N_i})'$ is reduced by randomly selecting subsets $X_i^* = (x_{i1}^*, x_{i2}^*, \dots, x_{kt}^*)'$ where $k < N$. For each X_i^*, bagging is used as the forecasting method by averaging the forecasts $B^{-1}\sum_{b=1}^{B}\hat{y}_{t+h}^{(b)}$, where $\hat{y}_{t+h}^{(b)}$ is computed using a bootstrap sample $X_t^{*(b)}$ and $y_{t+b}^{(b)}$. This procedure is repeated for multiple subsets, and the forecast average is computed for each subset. By subsampling, the correlation is reduced as it results in regression trees with different structures, leading to more stable forecasts. In our forecast exercise we set the subset of predictor variables at $k = N/2$.

3.3 Feedforward neural networks

In this paper we use feedforward neural networks (FFNN) described in complete detail by LeCun et al. (2015). For a complex overview of neural network theory please refer to the same paper. FFNN are a type of forecasting model with a single hidden layer written as

$$y_{t+h} = f(X_t) + \varepsilon_{t+h}, \tag{5}$$

where

$$f(X_t) = \sum_{j=1}^{q} \theta_j \sigma\left(w_j' X_t + b_j\right) + b, \tag{6}$$

Here, σ represents an activation function, q denotes the number of hidden units (neurons), and ε_{t+h} is the forecast error. We estimate the parameters $\left(\theta_j, w_j', b_j, b\right)$ for $j = 1, \dots, q$ by minimizing the least square criterion.

Two common activation functions considered in this context are the sigmoid function written as

$$\sigma(z) = \frac{1}{1 + e^{-z}} \tag{7}$$

and the ReLU (rectified linear unit) function:

$$\sigma(z) = \begin{cases} 0 & \text{if } z < 0 \\ z & \text{otherwise} \end{cases}, \tag{8}$$

where z represents the input of a hidden layer. These activation functions help introduce nonlinearity and allow the FFNN model to capture complex relationships in the data. In this paper we specify FFNNs according to Masters (1993) and Gu et al. (2019). Their hidden layers range from 1 to 5 with the number of a hidden unit q_ℓ in each layer ℓ following the geometric pyramid rule of Masters (1993).

3.4 Accuracy and forecast combination

In this paper we use standard metrics to measure the forecast accuracy of the models. We use the mean absolute error (MAE), written as

$$\mathrm{MAE} = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n} = \frac{\sum_{i=1}^{n} |e_i|}{n}, \tag{9}$$

which represents the absolute value of the difference between forecasted values and observed values. The next measurement is the root-mean-square error, written as

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{N} (x_i - \hat{x}_i)^2}{N}} \tag{10}$$

which gives us the quadratic average of the differences between the forecasted and observed values. For a detailed description of these measures please see Clements and Hendry (2011).

To continue with, we explain our choice to include the mean and trimmed mean and compare them to our forecasts. Clements and Hendry (2011) states that the combination of individual forecasting models can substantially improve forecast accuracy. Burgi (2015), for example, argues that it is hard to achieve more accurate forecasts than by combining the individual models using equal weights. Because of this, we first combine the models by taking the arithmetic mean of the individual forecasts. In addition, Armstrong (2001) argues that a trimmed mean may also improve forecast accuracy. To calculate the trimmed mean, we exclude the lowest and highest forecast values for each single observation. We then calculate the arithmetic mean of the remaining forecasts.

4 Data description

We used a dataset of 137 seven monthly observations of 37 macroeconomic variables from Slovakia, resulting in 5069 data points (N = 37, T = 137, NxT = 5069). To avoid the disruptive effects of COVID, the data spans from November 2008 to December 2019. We obtained the data from the database of the National Bank of Slovakia. The dataset included times series of industrial production by sector, employment by sector, wages, inflation, interest rate, electricity consumption, overall energy consumption, investment and new vehicles registered. Given the problems with data availability and relatively short time series, we did one-period-ahead forecasts of industrial production, inflation and unemployment.

To begin with, we cleaned the dataset by removing any missing values (NA) and stationarized the time series. Then, we divided the data into two subsamples for forecasting purposes. The first subsample consisted of 70% of all observations across all predictors and was designated as the training dataset. The second subsample encompassed 30% of the total observations and served as the testing dataset. Once the data is divided, we proceed to do one-period-ahead forecasts using the models described in section 3.

5 Results

In this section we present our results. We began our analysis by estimating a simple linear regression model, which serves as our benchmark. The MAE of this model is 0.1077 for industrial production, 0.0099 for inflation and 0.0188 for unemployment, which can be considered fairly low.

5.1 Performance of ensemble machine learning methods

The results of our forecasts using ensemble machine learning methods are in Table 1 for MAE and Table 2 for RMSE. Our first ensemble model is a single regression tree, which is inappropriate on its own, but can showcase how much improvement bagging, random forests and boosting provides compared to it. The MAE of this model is 0.1025 for industrial production, 0.0104 for inflation and 0.0171 for unemployment. This means that despite being fairly simple, a single regression tree is capable of outperforming our benchmark in the case of industrial production and unemployment.

We employ bagging, described above, as the first ensemble machine learning method. We use 13 bootstrap samples and 500 trees for each forecasted variable, which allows us to accommodate the relatively large dataset. The MAE of this model is 0.0843 for industrial production, 0.0063 for inflation and 0.0159 for unemployment This means that bagging improves the accuracy of a single tree model and outperforms the linear model for each of the forecasted variables.

Our next method of choice is the random forest. We are interested in confirming whether all ensemble machine learning methods overperform or it is just bagging. The description of random forests is provided in section 3. The number of variables randomly sampled as candidates at each fit is 4, while the number of trees remains the same. The MAE of this model is 0.0838 for industrial production, 0.0064 for inflation and 0.0145 for unemployment. This means that the random forest model also outperforms both the single tree and benchmark models for all the variables, while outperforming the bagged model as well for industrial production and unemployment, respectively.

Boosting is the last of the ensemble methods. Based on the method's description we set the number of trees higher to 5000. The MAE of this model is 0.0990 for industrial production, 0.0072 for inflation and 0.0165 for unemployment. This means that it still outperforms both the benchmark and single tree models, but it is less accurate than the other two ensemble machine learning methods.

To sum up, all ensemble machine learning methods outperform our benchmark model. Boosting is generally less accurate in our case than bagging and the random forest model, and the bagged model outperforms the random forest for industrial production and unemployment, while being slightly less accurate when forecasting inflation. The RMSEs of the forecasts, shown in Table 2, support our findings.

Table 1: MAE of ensemble machine learning methods

	Industrial production	Inflation	Unemployment
Linear regression	0.10771261	0.009981975	0.01883029
Single Tree	0.10255418	0.010446749	0.01710079
Bagging	0.08431572	0.006397466	0.01596954
Random Forest	0.08381620	0.006477983	0.01456082
Boosting	0.09909545	0.007296080	0.01658155

Source: author's forecasts based on NBS data

Table 2: RMSE of ensemble machine learning methods

	Industrial production	Inflation	Unemployment
Linear regression	0.1470913	0.012528327	0.03243916
Single Tree	0.1266006	0.012671102	0.03035339
Bagging	0.1028408	0.007980597	0.02973794
Random Forest	0.1021020	0.008131847	0.02957269
Boosting	0.1255822	0.008518829	0.02923006

Source: author's forecasts based on NBS data

5.2 Performance of regularized least squares methods

In this subsection we assess the forecast performance of the regularized least squares methods. The MAE of these methods is in Table 3 while the RMSE is in Table 4. We compare these results to the same benchmark. We begin with the ridge regression, which is the shrinkage method. Our first aim is to find the optimal value of λ, which we do according to the literature standard by employing 5-fold cross-validation. The values of the hyperparameters are calculated for each forecast step. For example, one optimal value of λ in our case is 0.003981072. The ridge model has a MAE of 0.8380 for industrial production, 0.0070 for inflation and 0.0141 for unemployment. This means that the ridge model outperforms the benchmark.

Lasso is our second regularized least squares method. The hyperparameter value is computed the same way. The lasso model has a MAE of 0.8381 for industrial production, 0.0071 for inflation and 0.0146 for unemployment. This means that there is a miniscule difference between the accuracy of the Ridge and Lasso models, but this difference is statistically insignificant.

The last regularized least squares method is the elastic net regression, where we compute the optimal hyperparameter values the same way. For example, optimal λ is 0.001209406 and

optimal ω is 0.06185348. The elastic net model has a MAE of 0.8381 for industrial production, 0.0071 for inflation and 0.0147 for unemployment.

Table 3: MAE of regularized least squares methods

	Industrial production	Inflation	Unemployment
Linear regression	0.10771261	0.009981975	0.01883029
Ridge	0.08380991	0.007059291	0.01418218
Lasso	0.08381435	0.007199378	0.01465475
Elastic net	0.08381435	0.007103715	0.01477725

Source: author's forecasts based on NBS data

In conclusion, all three models outperform our benchmark linear regression model. The performance of the models is almost completely identical. The RMSE of the models, shown in Table 4, supports our findings.

Table 4: RMSE of regularized least squares methods

	Industrial production	Inflation	Unemployment
Linear regression	0.1470913	0.012528327	0.03243916
Ridge	0.1016013	0.008925419	0.02950428
Lasso	0.1016086	0.009050781	0.02977501
Elastic net	0.1016086	0.008995676	0.02952099

Source: author's forecasts based on NBS data

5.3 Performance of neural networks and results summary

In this subsection we present the MAE and RMSE of our neural network model. In addition, we also compare ensemble machine learning and regularized least squares methods with each other and compare the performance of individual forecast models to the mean and trimmed mean of individual forecasts. Table 5 contains the MAE for all models and Table 6 shows the RMSE.

To begin with, the MAE of our feedforward neural network model is 0.0924 for industrial production, 0.0085 for inflation and 0.0174 for unemployment. This means that even though it outperforms both our benchmark and linear regression models, it is generally the least accurate from all the other machine learning techniques, except for boosting, which is less accurate only for industrial production.

Comparing ensemble machine learning and regularized least squares methods, we see that their performance is fairly similar, so we cannot generalize and conclude that one method is better

than the other. However, the aim of this paper is not to establish primacy, but to find out whether machine learning methods can outperform a benchmark standard econometric model in our short-term forecasting exercise. We conclude that our aim was fulfilled and that machine learning methods proved themselves to be more accurate for Slovakia if we have a relatively large dataset. This might suggest that there are, for example, nonlinearities in the available dataset.

Our secondary aim was to compare the performance of individual forecasting models to the mean and trimmed mean. The simple mean of the forecasts is 0.09126 for industrial production, 0.0078 for inflation and 0.0160 for unemployment. Our findings are interesting. As mentioned in section 3, it is really rare to beat even the simple mean of individual forecasts. In our case, however, every artificial intelligence method beats the simple mean, except for the FFNN model. This means that our individual models are more accurate than their equally weighted combination.

Given this unusual, but possible and empirically documented situation, we aim to make the mean more accurate by "trimming" out 15% of the models. The trimmed mean of industrial production is 0.0899, for inflation is 0.0076 and for unemployment 0.0158. This means that even though there is some improvement over the equally weighted average, the trimmed mean still underperforms the individual forecasts of all the machine learning models, except for boosting, which has a higher MAE for industrial production.

Table 5: MAE of all the methods

	Industrial production	Inflation	Unemployment
Linear regression	0.10771261	0.009981975	0.01883029
Single Tree	0.10255418	0.010446749	0.01710079
Bagging	0.08431572	0.006397466	0.01596954
Random Forest	0.08381620	0.006477983	0.01456082
Boosting	0.09909545	0.007296080	0.01658155
Ridge	0.08380991	0.007059291	0.01418218
Lasso	0.08381435	0.007199378	0.01465475
Elastic net	0.08381435	0.007103715	0.01477725
FFNN	0.09243234	0.008594091	0.01747695
Mean	0.09126279	0.007839636	0.01601490
Trimmed mean	0.08997751	0.007673216	0.01587452

Source: author's forecasts based on NBS data

Given this interesting finding, our next aim is to find different weighting procedures which outperform the individual models. This topic is going to be the subject of further research.

Table 6: RMSE of all the methods

	Industrial production	Inflation	Unemployment
Linear regression	0.1470913	0.012528327	0.03243916
Single Tree	0.1266006	0.012671102	0.03035339
Bagging	0.1028408	0.007980597	0.02973794
Random Forest	0.1021020	0.008131847	0.02957269
Boosting	0.1255822	0.008518829	0.02923006
Ridge	0.1016013	0.008925419	0.02950428
Lasso	0.1016086	0.009050781	0.02977501
Elastic net	0.1016086	0.008995676	0.02952099
FFNN	0.1316060	0.010962174	0.02978506
Mean	0.1156268	0.009751639	0.02999096
Trimmed mean	0.1131355	0.009587579	0.02974991

Source: author's forecasts based on NBS data

6 Conclusion

In this paper we forecasted industrial production, inflation and unemployment for Slovakia using machine learning methods applied to a relatively large dataset. Our results are appealing. Firstly, every machine learning category, namely ensemble machine learning, regularized least squares and neural network methods outperformed the benchmark model. This finding was supported by both the MAE and RMSE metrics. Secondly, we conclude that we also achieved our secondary aim. We compared the forecast performance of individual machine learning models to their arithmetic mean and trimmed mean. We found that individual models can outperform their averages, which is a relatively rare occurrence.

To continue our research, we would like to conduct the same exercise for the rest of the Visegrad Four economies, starting with the Czech Republic. In addition, given the importance of forecast combination, we are going to test more complex weighting procedures to find the optimal way to weight the forecasts of our machine learning models to increase accuracy even more.

7 Acknowledgement

This research was financially supported by VEGA No. 1/0047/23 The importance of spatial spillover effects in the context of the EU's greener and carbon-free Europe priority.

8 References

ARMSTRONG, J. S. (2001). Combining Forecasts. *University of Pennsylvania Scholarly Commons Marketing Papers 34*, s. 1-21.

BALDACCI, E.; BIJONI, D.; KAPETANIOS, G.; KRISCHE, S.; MARCELLINO, M.; MAZZI, G.L. and PAPAILIAS, F. (2016). Big Data and Macroeconomic Nowcasting: from data access to modelling. 2016 vyd. Luxembourg: Publication Office of European Union, 2016.

BREIMAN, L. (1996). Bagging predictors. Machine Learning. 1996, Vol. 24, No. 2, s. 123–140.

BREIMAN, L. (2001). Random forest. Machine Learning. 2001, Vol. 45, No. 1, s. 5–32.

BREIMAN, L.; FRIEDMAN, J.; STONE, C.J. and OLSHEN, R.A. (1984). Classification and Regression Trees. Chapman and Hall/CRC, New York, NY. 1984.

BÜHLMANN, P. and YU, B. (2002). Analyzing bagging. *Annals of Statistics*. 2002, Vol. 30, No. 4, s. 927–961.

BÜRGI, C. (2015). Can a subset of forecasters beat the simple average in the SPF? *RPF Working Paper No. 2015-001.*

CLEMENTS, M. P. and HENDRY, D.F. (2011). *The Oxford Handbook of Economic Forecasting*. 1. vyd. Oxford: Oxford University Press, 2011.

FRIEDMAN, J.; HASTIE, T., and TIBSHIRANI, R. (2001). The elements of statistical learning. 1. vyd. New York: Springer, 2001.

GU, S.; KELLY, B. and XIU, D. (2019). Empirical asset pricing via machine learning. NBER Working Paper 25398. 2019, s. 1-46.

GREEN, W. H. (2003). Econometric Analysis. 5. vyd. New Jersey: Pearson Education Inc., 2003.

HASTIE, T.; TIBSHIRANI, R. and FRIEDMAN, J. (2009). The Elements of Statistical Learning. 2. vyd. New York: Springer, 2009.

HOERL, A.E. and KENNARD, R.W. (1970). Ridge regression: biased estimation for nonorthogonal problems. Technometrics. 1970, Vol. 12, No. 1, s. 55–67.

INOUE, A. and KILIAN, L. (2008). How useful is bagging in forecasting economic time series? A case study of U.S. consumer price inflation. Journal of American Statistical Association. 2008, Vol. 103, No. 482, s. 511–522.

JAMES, G.; WITTEN, D.; HASTIE, T. and TIBSHIRANI, R. (2013). An introduction to statistical learning: with Applications in R. 1. vyd. New York: Springer, 2013.

KURIHARA, Y. and FUKUSHIMA, A. (2019). AR Model or Machine learning for Forecasting GDP and Consumer Price for G7 countries. *Applied Economics and Finance*. 2019. Vol. 6, No. 3, s. 1-7.

LECUN, Y.; BENGIO, Y. and HILTON, G. (2015). Deep learning. Nature. 2015, Vol. 521, s. 436-444.

MAEHASHI, K. and SHINTANI, M. (2020). Macroeconomic forecasting using factor models and machine learning: an application to Japan. Journal of the Japanese and International Economies. 2020, Vol. 58, s. 1-17.

MASTERS, T. (1993). Practical Neural Network Recipes in C++. 1. vyd. Sand Diego: Academic Press, 1993.

MEDEIROS, M.C.; VASCONCELOS, G.F.R.; VEIGA, Á. and ZILBERMAN, E. (2019). Forecasting Inflation in a Data-Rich Environment: The Benefits of Machine Learning Methods. *Journal of Business & Economic Statistics*. 2019. Vol. 39, No. 1. s. 98-119.

ÖZGÜR, Ö. and AKKOC, U. (2021). Inflation forecasting in an emerging economy: selecting variables with machine learning algorithms. *International Journal of Emerging Markets.* 2021. Vol. 16, No. 2, s. 1-21.

RICHARDSON, A.; van FLORENSTEIN MULDER, T. and VEHBI, T. (2018). Nowcasting New Zealand GDP using Machine Learning Algorithms. *CAMA Working Paper 47/2018*. 2018. s. 1-16.

SCHAPIRE, R.E. (1990). The strength of weak learnability. Machine Learning. 1990, Vol. 5, No. 2, s. 197–227.

TIBSHIRANI, R. (1996). Regression shrinkage and selection via lasso. Journal of the Royal Statistical Society: Series B. 1996, Vol. 58, No. 1, s. 267–288.

ZOU, H. and HASTIE, T. (2005). Regularization and variable selection via the elastic net. Journal of the Royal Statistical Society: Series B. 2005, Vol. 67, No. 2, s. 301–320.