ASECU South-Eastern Europe Journal of Economics 1 (2020) 39-99 SEEJE

FORECASTING ECONOMIC RECESSIONS USING MACHINE LEARNING: AN EMPIRICAL STUDY IN SIX COUNTRIES

ANDREAS PSIMOPOULOS*
ETH Zurich, Switzerland

Abstract

This paper proposes a methodology for forecasting economic recessions using Machine Learning algorithms. Among the methods examined are Artificial Neural Networks (ANN), Support Vector Machines (SVM) and Random Forests. The datasets analysed refer to six countries (Australia, Germany, Japan, Mexico, UK, USA) and cover a time span of more than 40 years. All methods are compared against each other in terms of six evaluation metrics on their out-of-sample performance. In contrast to most similar empirical studies, the methodology developed focuses on the timepoints of the last four quarters before a recession begins rather than on those of a recession *per se*. It has been found that the SVM method tends to outperform the others, as it classified correctly at least 75% of the pre-recessionary periods for half of the countries, with mean overall classification accuracy around 90% in these cases. Moreover, for all the countries under study, the traditional Logit and Probit models are always inferior to at least one Machine Learning-based model. Additionally, it turns out that macroeconomic variables representing a kind of debt – such as, household debt – are most frequently considered as important across the six datasets, in terms of the Mean Decrease Gini measure.

JEL Classification: C18, C45, C53, E37
Keywords: Forecasting recessions, Machine Learning-based Econometrics, Gini importance, Support Vector Machines

Acknowledgements: I would like to thank Prof. Dr. Nicolai Meinshausen for his valuable assistance throughout all phases of this project. I would also like to thank the two anonymous referees for their constructive comments and suggestions.

**Corresponding Address:* Andreas Psimopoulos, ETH Zurich, Department of Mathematics, Rämistrasse 101, 8092 Zurich, Switzerland. E-mail: andreas.psimopoulos@alumni.ethz.ch

Introduction

Modern global economy is a highly complex dynamic system. Richard M. Goodwin (1951) highlighted the importance of incorporating nonlinear differential or difference equations in the analysis of business cycles, instead of following oversimplified linear approaches. Wong *et al.* (2011, p. 432) argue that the global financial system is becoming more and more complex, as interconnectivity among financial systems, markets and institutions increases. Moreover, Barnett *et al.* (2015, p. 1750) argue that an alternative explanation for the existence of business cycles is the chaotic nature of economic systems. According to the latter explanation, business cycles are not caused by exogenous shocks, as the popular opinion holds, but they are created endogenously due to the stochastic behaviour of economic systems. Nevertheless, the authors conclude that, currently, it is not feasible to verify whether this chaotic behaviour has its roots in internal factors or not. One of the biggest challenges in Economics is the accurate prediction of some measures of interest, such as the Gross Domestic Product (GDP), for various reasons, e.g., policy making, financial speculation, etc. However, long-term predictions in chaotic systems are impossible, due to the inherent property of systems' sensitive dependence on initial conditions. Since it is not clear whether economic systems are truly chaotic systems or not, the objective of the present paper is to propose a Machine Learning-based methodology, the goal of which is to provide reliable *short-term* predictions of economic recessions. The methodology proposed focuses on the signs that precede significant downturns of economic activity. In other words, our goal is to capture the dynamics of some important macroeconomic factors before a recession occurs, in order to use these signs as indicators for upcoming recessions. The potential benefit is that such predictions can be taken into account by policymakers, giving them the chance to design and apply more effective policies.

According to the International Monetary Fund, there is no official definition of the term *economic recession*. However, a practical definition that seems to be widely accepted is the following: "*Recession is a period of two consecutive quarters of decline in a country's real GDP*" (Claessens & Kose, 2009, p. 52). This definition is also accepted in the framework of this paper. A special case of recessions are the so-called *depressions*. A depression is a severe and long-lasting recession (Hall & Lieberman, 2013, p. 125). Although there is no general consensus regarding the magnitude and the duration that labels a recession as depression, most analysts make this distinction if the decline in real GDP exceeds 10% (Claessens & Kose, 2009, p. 53). Generally speaking, depressions are very rare, and, thus, we do not study them separately in the models presented below.

The capacity to predict economic recessions is not the only open research question regarding them. It is a fact that there has been a two-hundred-year debate in Economics about what causes recessions and depressions – and there has been no general agreement so far (Knoop, 2015, p. 4). Therefore, by looking for macroeconomic signs

before such events, it may be possible to confirm an existing theory about why recessions happen or pave the way for a new one. At this point, it should be clarified that there is no clear distinction between statistical and machine learning methods. They, rather, form a spectrum of different methods with similar goals. Machine learning is a field of Computer Science with sound statistical foundations and Statistics is a branch of Mathematics which is increasingly taking more advantage of algorithms and computational infrastructure. A part of this spectrum is presented in the discussion about methods that can be used for solving our main problem.

The paper commences with a review of the literature about theories and recent findings relevant to economic recessions, focusing on topics related to forecasting. In the next section, the methodology applied is presented, which is followed by the section of corresponding results. The fifth section includes the discussion of results, while the paper is summarised with the main conclusions along with reference to some topics for potential further research. An appendix can be found at the end of the paper, which provides additional details for a variety of topics mentioned in the main text.

Literature Review

A short review of theories relevant to economic recessions

As already mentioned, what causes recessions is an open research question. There is a plethora of schools of economic thought, simply because there is no global consensus on how economies operate. These schools of thought generally build their theories on different axioms and it is likely that two schools may have starkly different opinions about a topic. The existence of economic recessions is such a topic that one can find a lot of different explanations in literature about why they occur. For us to find if any such theory can be empirically verified, we have used several variables in our models, which arise from theories related to economic recessions. In this subsection we briefly present these theories.

It is well known that, during recessionary periods, a characteristic situation in the economy is low profitability of the firms. Adam Smith (1723–1790), a social philosopher considered to be the father of Classical Economics, mentions three reasons that cause low profitability: (a) competition in the labour market, which leads to higher wages, and, therefore, decreased profits; (b) competition in the capital market, which leads to higher prices of capital goods, and (c) competition in the consumer goods market, which forces capitalists to sell at cheaper prices, which also diminishes profits (Smith, 1776 [1977], pp. 129, 469). These reasons are linked to macroeconomic variables like unemployment or inflation, which may be found useful for the models of this paper. Another influential classical economist, David Ricardo (1772–1823), stressed the fact of the negative economic and social consequences that arise due to the endlessly growing population (Ricardo, 1821 [2001], pp. 59-64). Hence, it may

be useful to also include demographic variables in a model intended for predicting recessions, which is what we have done, as presented below. Karl Marx (1818–1883) provides a theoretical framework that specifies the exact point at which an economic crisis erupts, which is the onset of the phenomenon we are interested in. Marx argues that periodical depreciation of existing capital is associated with crises in the production process. The birth of such crises occurs at the point of *absolute over-accumulation of capital* (Marx, 1894 [2010], pp. 176-178). A mathematical explanation for this concept can be found in Tsoulfidis (2010, pp. 119-120). According to his analysis, the absolute over-accumulation of capital happens when the elasticity of profit rate with respect to capital (denoted as $e_{r,C} = \frac{dr}{dC}\frac{C}{r}$) is –1. This suggests that $e_{r,c}$ is likely to be a good predictor of economic crises and – therefore – recessions.

John Maynard Keynes (1883–1946) – one of the most influential economists of the 20th century – was very critical of the Classical model ideas about business cycles. One of the basic assumptions of the Classical model is that perfect competition exists in all markets, which always leads them in equilibrium. In this model, business cycles do not exist; recessions happen due to government policies and regulations (Knoop, 2015, pp. 40, 44). Keynes' explanation about why recessions happen points to a new – for our analysis – variable: expectations about future earnings (Keynes, 1936 [2013], pp. 46-47). He argues that changes in expectations gradually produce similar oscillations in employment and what mainly determines expectations – especially short-term ones – is the most recent actual results (Keynes, 1936 [2013], pp. 49-51). Hence, qualitative indicators, such as the Business Confidence Index (BCI)[1], might be used for incorporating these aspects in a statistical or a machine learning model. Keynes also referred to the concept of *paradox of thrift*, according to which, every attempt to increase aggregate saving – at the expense of consumption – is necessarily self-defeating (Keynes, 1936 [2013], pp. 83-84). In this framework, higher savings, at the expense of consumption, reduce aggregate demand and, thus, cause production to fall. So, a recession may begin – or may be prolonged – after an increase in aggregate saving, even if such a sign seems good at first sight. Therefore, aggregate saving and consumption are potential predictors of economic recessions, among others. What Keynes proposes for recovering from a recession is that the government should intervene in the economy with expansionary fiscal policies (Knoop, 2015, pp. 56-57).

Milton Friedman (1912–2006) was the founder of the School of *Monetarism*. For reasons that are out of the scope of this paper, Keynes believed that only fiscal policies can be effective[2]. Monetarists are sceptical about such a view. Evidence from the

1. "The business confidence index (BCI) is based on enterprises' assessment of production, orders and stocks, as well as their current position and expectations for the immediate future. Opinions compared to a 'normal' state are collected and the difference between positive and negative answers provides a qualitative index on economic conditions." (OECD, 2018a).
2. A summary of Keynes' justification about this belief can be found in Knoop (2015, pp. 55-57).

economic history of the USA suggests that there is strong correlation between changes in money stock and business cycles. Friedman & Schwartz (1963, pp. 676-695) argue that this correlation can be verified for all U.S. recessions during the 1867–1960 period. Their concluding remark is that changes in money supply play the major role in the formation of business cycles and a less important role in short-term fluctuations of economic activity. Friedman (1968, p. 17) suggests that economic stability can be achieved by setting steady but moderate growth in the quantity of money. According to all these ideas from the Monetarist model, one can say that money supply is, potentially, a good predictor of economic recessions, given that there is a causal relationship from changes in money stock to business cycles.

Irving Fisher (1867–1947) concluded that the two factors with prevailing impact on the evolution of business cycles are over-indebtedness and deflation (Fisher, 1933, p. 341). According to Fisher, debt and price levels are primary variables when studying business cycles, in the sense that other similarly important variables are affected by them. Speaking in a business-related context, firms experience a loss in their profits, due to lower prices (deflation); employment, output and trade are reduced (recession), some of the firms go bankrupt, and pessimism, along with loss of confidence, lead to more money-saving and fewer transactions (Fisher, 1933, p. 342). If this theory is confirmed by data, then what we need to find are those values of deflation and private sector debt that signal the occurrence of an upcoming recession.

Joseph Schumpeter (1883–1950) was one of the most important advocates of *Austrian Economics*. Schumpeter emphasises the evolutionary character of the capitalist process. By using the term *creative destruction,* he puts forward an alternative explanation regarding the existence of business cycles (Schumpeter, 1942 [1994], pp. 81-86): As some firms embody innovation (*creation* of new structures), they become able to produce more and sell at cheaper prices in the long-run, leading their competitors to change or to leave the market (*destruction* of existing structures). It is easily conceivable that this innovation dynamics influences business cycles. Hence, innovation indicators could be useful variables for further analysis, because it may be the case that such an indicator follows a specific pattern before the onset of a – rather prolonged – recession.

Finally, we conclude this subsection with the models of *New Keynesian Economics*. These models were developed in 1980s, as a response to the criticism of Keynesian Economics. Regarding what can cause a recession in the framework of New Keynesian Economics, there are three explanations (Knoop, 2015, p. 136): a) change in expectations (old Keynesian approach); b) contraction in money supply (Monetarist approach); c) increase in default risk perceptions. As we can realise, the new variable for our analysis lies in the third explanation. Speaking in a countrywide context, default risk is one of the factors that influence long-term interest rates. According to the OECD (2018b), these interest rates are determined by the amount charged by the

lender, the risk the borrower undertakes and the fall in capital value. If it is perceived that a country will face difficulties in paying its debt obligations promptly, long-term interest rates imposed on it are increased due to default risk. Consequently, business investment falls. Therefore, one additional potential predictor of economic recessions is long-term interest rates.

Recent studies focusing on prediction of economic recessions

In this part we present a review of recent studies focused on the topic of forecasting recessions. Estrella & Mishkin (1998) investigated several leading indicators for the prediction of U.S. recessions, such as stock prices, interest rates, etc. In order to estimate the probability of the occurrence of a recession, these authors used a Probit model. Their analysis was focused on the out-of-sample performance of their models, up to eight quarters ahead, and they found that the best predictors of U.S. recessions were stock prices and the yield curve spread[3]. Chauvet & Potter (2005) considered an extended Probit specification. In particular, their work is based on a dynamic Probit framework, where dependent variables are regressed on their lagged values and other exogenous regressors, namely, yield spreads. Their best model allowed for multiple breakpoints across business cycles and autocorrelated errors and it achieved better in-sample fit than the model by Estrella & Mishkin. Christiansen (2013) used a Probit model in order to examine the forecasting ability of yield curve spreads in simultaneous recessions of six countries (Australia, Canada, Germany, Japan, United Kingdom and United States). She considers a recession as 'simultaneous' if it occurs in at least half of the countries studied. She found that, at short horizons, only the German yield spread was significant in explaining future simultaneous recessions, but, at long horizons, both U.S. and German spreads were such.

Dovern & Huber (2015) found that the Global Vector Autoregressive (GVAR) approach produced more accurate predictions of recessions than country-specific time series models. The authors defined a period of at least two consecutive quarters with declining GDP as recession, and they used a dummy variable (binary indicator) to encode this in their data. They investigated 36 countries over a period of ten years[4] and their goal was to provide good probability forecasts for the occurrence of a recession, within a Bayesian framework. Using the real GDP, the change of CPI, the real equity prices, the real exchange rate and the short/long-term interest rates as variables, they successfully constructed a GVAR model that outperformed both of the two benchmark models[5] in forecasting accuracy for the majority of the countries.

3. A yield curve with a negative slope is often considered as a sign of an upcoming recession. The negative slope of the yield curve means that the interest rate spread is negative.
4. They used quarterly data. The dataset contained data from 1979.Q2 (i.e., the second quarter of 1979) to 2013.Q4. For the verification period, they used 40 observations from 2004.Q1 to 2013.Q4.
5. These models were the Bayesian VAR (BVAR) and the Bayesian univariate autoregressions (AR).

Kauppi & Saikkonen (2008) found that dynamic Probit models outperform static ones in terms of both in-sample and out-of-sample predictions. The authors' goal was to build different forecasting models to predict U.S. recessions and to compare them to each other. They used the definition of recessions[6] by the National Bureau of Economic Research (NBER) and they also encoded recessionary periods with a binary variable (1: recession, 0: otherwise). They also used quarterly data referring to the 1955.Q4 – 2005.Q4 period and the best model of their analysis was only based on the interest rate spread and the binary variable.

We proceed with the paper by Gogas *et al.* (2015). To the best of that paper authors' knowledge, it was the first attempt to forecast GDP cycles using a Support Vector Machines (SVM) classifier[7] on data relevant to the yield curve. Gogas *et al.* found that both the short-term and the long-term interest rates had an important role in forecasting future recessions. They used quarterly data of the U.S. GDP and interest rates, from 1967.Q3 to 2011.Q4. Moreover, they used a definition for recessions that differed from the two mentioned previously: They considered every deviation of GDP *under* the long-run trend as a recessionary period. The best performing model was a radial kernel SVM, which achieved in-sample test accuracy of 73.3% and out-of-sample overall accuracy of 66.7%. Döpke *et al.* (2017) applied a machine learning approach known as *Boosted Regression Trees* (BRT). Their goal was to find the predictive value of several leading indicators for forecasting recessions in Germany. They used 35 leading indicators related to the German economy, which were collected on a monthly basis. Some of them were data about money supply, unemployment, price levels, exchange rates and data about interest rates. Their sample period was from 1973.M1 (i.e., January 1973) to 2014.M12. They did not use a specific definition for recessions; for their context, recessionary periods are those characterized as *troughs* by the Economic Cycle Research Institute (ECRI, 2013) and they encoded them in a binary variable. Regarding their findings, they provide evidence that the BRT approach has better out-of-sample performance in comparison to variants of the alternative Probit approach, and that the most influential leading indicators were: a) the short-term interest rate of money market instruments[8], and b) the spread *yield on ten-year government bonds* minus *money market rate*.

6. "The NBER does *not* define a recession in terms of two consecutive quarters of decline in real GDP. Rather, a recession is a significant decline in economic activity spread across the economy, lasting more than a few months, normally visible in real GDP, real income, employment, industrial production, and wholesale-retail sales." (NBER, 2010).
7. More details about the Machine Learning methods mentioned in this paper can be found in Appendix, A.1. A short description of the methods used is cited at the end of the next section.
8. Also known as *money market rate*.

Plakandaras *et al.* (2017) compared the performance of SVM models with that of dynamic Probit models in forecasting U.S. recessions. They used data from 1871.M1 up to 2016.M6 and, with regard to how the recessionary periods were specified, they used the definition of NBER, as Kauppi & Saikkonen (2008) did. Plakandaras *et al.* included in their analysis explanatory variables about stock prices, oil prices, financial indicators, money supply and the yield curve. They found that, for short-term predictions, Probit models outperformed SVM, but, for longer horizons, the latter provided more accurate forecasts. To be more specific, Probit models showed better out-of-sample performance in forecasting horizons of 1 and 3 months, while SVM performed better in forecasting horizons of 6, 12, 18, 24 and 36 months. Lastly, we complete this section with the paper by Kiani (2008). The goal of this paper was to apply Artificial Neural Networks (ANN) on forecasting recessions for a set of countries. These countries were Canada, France, Germany, Italy, Japan, the UK and the USA. Kiani's paper provides some first insights about the capabilities of ANN in forecasting recessions. The data used for this research were quarterly for all countries, from 1965.Q1 to 2004.Q4. Variables referred to money supply, stock price indices, several interest rates and others. The author investigated ten models: one for each of the seven variables used and three models based on variable combinations. Regarding the results, each country had a different set of candidate predictors of recessions, according to the out-of-sample forecasting performance of the models employed. The most common ones were the stock price indices and the spread between bank rates and risk-free (i.e., T-Bill equivalent for all countries) rates. Regarding forecasting accuracy, the author defined a new metric that considers both Type I and Type II errors; according to his definition, missed recessionary periods and other missed periods, respectively. In this metric (Kiani, 2008, p. 4), forecasting accuracy exceeded 80% for most of the countries, which is noteworthy. It seems that the flexibility of ANN makes it possible to sufficiently capture the nonlinear features of business cycles, regardless of the country. It would be interesting to see, though, how well these models score in other metrics, too.

Methodology

In this section, we present all methodological issues concerning the models developed. In order to better evaluate the performance of the presented methodology, the author's decision was to apply it for the datasets of six countries, namely: Australia (AUS), Germany (GER), Japan (JAP), Mexico (MEX), the United Kingdom (UK), and the United States of America (USA). There is no objective, strictly defined criterion for this selection, but the selection was not random, either. The main rationale was to choose countries from different continents that are likely to differ from each other in economic terms. If results from heterogeneous countries converge, this is an indication that the pre-recessionary conditions are common across countries and the corresponding model tends to generalise well. If they do not, one could associate pre-recessionary conditions with country-specific characteristics. Additionally, since

the Organization for Economic Co-operation and Development (OECD) provides a comprehensive database of economic and other factors, a prerequisite condition was that the countries to be selected should be OECD's members, in order to take full advantage of the database of the Organisation for the purposes of this paper[9].

Table 1 contains all the main variables of the datasets constructed. These variables are called 'main', in the sense that all other variables of the final datasets are some transformations of the former. The initial goal was to have each variable in quarterly frequency, from 1969.Q1 up to 2017.Q4 (196 possible observations). As GDP is at the centre of our attention in the context of this paper, this decision was made because the highest available frequency of GDP-related data was on a quarterly basis. However, some variables were provided only in yearly frequency and, for some countries, many variables had their first observation some time during 1990s. The consequence of these circumstances was that there were a lot of missing values in the six datasets, which, in turn, added an extra challenge during the pre-processing steps[10].

At this point it is important to make some remarks regarding the data downloaded. First, variable *GFCF* refers only to domestic investment. The OECD provides a variable for Foreign Direct Investment (FDI), but it is not included in the datasets because *no more* than thirteen observations were available for any country (i.e., a large number of values were missing). A similar situation came up during the collection of Mexico's *BankRate* data. The central bank of Mexico provides relevant data only from 2008 and on. Therefore, Mexico's dataset does not contain the *BankRate* variable. Regarding variable *M*1, the decision was to exclude it form Germany's dataset, because, since the establishment of the Eurozone, *M*1 has been the same for all its member-states.

With regard to variable *PPP*, as already mentioned in Table 1, it is measured in *national currency units/US dollar*. This means that for the USA this variable is always equal to one, which is a problematic situation if we want to keep it in the USA dataset. By looking at *PPP*, one essentially compares a country's cost of living with that of the USA. This can be considered as a price ratio between the two countries, where each price refers to a basket of goods and services. So, the question was what to do if we want to measure *PPP* for the USA. The idea here was to reverse the concept. Apart from the time series of separate countries, the OECD also provides *PPP* data for EU28; this means one additional time series, weighted across the 28 countries of the European Union. This time series was also expressed in relation to the U.S. dollar. Therefore, the idea here was to invert the PPP_{EU28} in order to express USA *PPP* in a hypothetical common currency of EU28 (i.e., *US dollars*/1*unit of "EU*28"). The ideal situation would be to have a time series weighted across *all* countries except for the USA, but inverting that of EU28 also seems a good approximation[11].

9. References regarding the variables downloaded can be found in Appendix, A.2.
10. A detailed presentation of the pre-processing steps can be found in Appendix, A.3.
11. In OECD's database, no other weighted *PPP* time series consisted of more than 28 countries. In addition to that, the fact that the EU is in aggregate of the largest economies worldwide makes the choice of these 28 countries even more suitable for our purpose.

Variable name	Short description	Unit	Linked to a specific theoretical concept or paper
BankRate	Bank rate; interest rate at which the national central bank lends money to domestic banks.	% of principal	It was introduced in order to find if it has similar predictive importance to other types of interest rates.
BCI	Business Confidence Index; enterprises' expectations for the immediate future.	–	Expectations about future earnings, *John M. Keynes*.
CPI	Consumer Price Index; an index of the price level of consumer goods.	–	3rd argument about firms' low profitability, *A. Smith*. Also, part of *Fisher*'s debt-deflation theory.
Fdebt	Financial corporations' debt to equity ratio; a measure of corporations' debt.	–	Debt-deflation theory, *Irving Fisher*.
Fprof	Net operating surplus of financial corporations; an indicator of the financial sector profitability.	% of net value added	Necessary for the construction of variable 'elasticity of profit rate with respect to capital', *K. Marx*; L. Tsoulfidis (2010).
Gdebt	General government debt-to-GDP ratio.	–	It was introduced because variables related to private sector's debt are also included.
GFCF	Gross Fixed Capital Formation; i.e. investment (domestic).	Growth rate	(generally used concept)
Gov	General government spending.	% of GDP	One of two main fiscal policy instruments, *J. M. Keynes*.
GrowGDP	Percentage change of real GDP.	–	(generally used concept)
HHC	Households' consumption expenditure.	% of GDP	Consumption shrinkage in the concept of 'paradox of thrift', *J.M. Keynes*.
HHdebt	Household debt.	% of net disposable income	Debt-deflation theory, *I. Fisher*.
LIR	Long-term interest rate; this can be considered a measure of default risk.	% of principal	Default risk is one of the potential causes of recession in the framework of *New Keynesian Economics*.
M1	An index of money supply (coins, banknotes and overnight deposits).	–	The core concept of Monetarism, *Milton Friedman*.
Mports	Trade in goods and services: Imports.	% of GDP	(generally used concept)
NFdebt	Non-Financial corporations' debt to surplus ratio; a measure of corporations' debt.	–	Debt-deflation theory, *I. Fisher*.
NFprof	Net operating surplus of non-financial corporations; an indicator of non-financial sector profitability.	% of net value added	Necessary for the construction of variable 'elasticity of profit rate with respect to capital', *K. Marx*; L. Tsoulfidis (2010).
Pop	Population.	10^6 persons	Threats by endlessly growing population, *David Ricardo*.
PPI	Producer Price Index; an index of the price level of producer goods.	–	2nd argument about firms' low profitability, *A. Smith*.
PPP	Purchasing Power Parities: the rates of currency conversion that equalize the purchasing power of different currencies by eliminating differences in price levels between countries.	National currency units / US dollar	Currency conversion (i.e., exchange rate) variables have been used in the papers by Dovern & Huber (2015) and Döpke et al. (2017).
RnD	Gross domestic spending on Research and Development; a measure of innovation.	% of GDP	Role of innovation in the concept of 'creative destruction', *Joseph Schumpeter*.
Sav	Saving rate.	% of GDP	Increase of savings in the 'paradox of thrift', *J.M. Keynes*.
SIR	Short-term interest rate. Monetary policy can be conducted by using it.	% of principal	Variable mentioned in the paper by Gogas et. al. (2015).
SPI	Share Price Index; another name for the already mentioned Stock Price Index.	–	A similar variable was mentioned in the paper by Kiani (2008).
Tax	Total tax revenue.	% of GDP	One of two main fiscal policy instruments, *J. M. Keynes*.
Unemp	Unemployment rate.	% of labor force	1st argument about firms' low profitability, *A. Smith*.
Wage	Average wage in total economy.	US dollars	1st argument about firms' low profitability, *A. Smith*.
Xports	Trade in goods and services: Exports.	% of GDP	(generally used concept)

Table 1. The main variables. References are presented in Appendix, A.2.

We close this section's introduction by mentioning the fact that certain variables were not at all available for some countries at the time of data collection, namely:

Table 2. List of missing variables per country.

Country	Missing variables
Australia	$Fprof, NFprof$
Japan	$PPI, Fprof, NFprof$
Mexico	$PPI, HHdebt, Gov$

The variable selection procedure

After the main pre-processing steps, we had six datasets with over a hundred variables in each one, and very few missing values at their top and/or bottom parts. The first question that came up before feeding the models with data was how to choose the explanatory variables. The goal was to have a relatively small number of well-chosen predictors, in order to build parsimonious models and provide reliable results. Thus, the idea was to exploit some beneficial outcomes of the Random Forests modelling. As seen in James *et al.* (2013, pp. 319-321), random forests are based on many different decision trees and, in turn, each decision tree is generally based on a different set of variables[12]. This fact gives us the possibility to measure a variable's importance in terms of how much an impurity measure – like the Residual Sum of Squares (RSS) or the Gini index – is decreased on average, for all the times this variable is selected as a predictor. This measurement can easily be made using the function varImpPlot() from package randomForest. The output of this function is a plot that one can use in order to assess importance of variables in a Random Forests model. For our case, since the target variable is binary, the measure that varImpPlot() uses is the Mean Decrease Gini (MDG). Consequently, the decision here was to fit a Random Forests model[13] and look at the output of varImpPlot() for the group of the most important variables in terms of MDG. This means that the number of initially selected variables varies from country to country and the selection procedure is mainly based on a soft (visual) rule. In other words, we are looking for a small group of variables that lie far from others in terms of MDG. This procedure is, in some sense, a competition among different theories and hypotheses about economic recessions based on real world data. If some variables are systematically characterised as important, this implies that the hypotheses they represent are close to reality. To the author's best knowledge, this is the first empirical study that compares the strength of all these hypotheses, which are briefly presented in the last column of Table 1.

12. Details regarding the methods mentioned can also be found in Algorithms 3 and 4 of the present paper.
13. Regarding the value of parameter mtry (number of variables sampled before each split) in randomForest(), the overall out-of-bag (OOB) error was taken into account by looking at the output of function plot.randomForest() (black line). To be specific, a value of mtry that quickly minimizes the OOB error was selected for each country, following a trial and error approach. Parameter ntree (number of trees to grow) was set at 10,000 in order to estimate each variable's importance to the best possible degree.

The Average Trees algorithm

A new algorithm was developed in order to identify macroeconomic conditions that precede recessions. The Average Trees algorithm can be considered as a robust version of Decision Trees and, specifically, a robust version of classification trees. The latter means that the target variable is a binary one, i.e., the *PreRecess*, which represents a pre-recessionary period[14]. The main idea of the methodology developed is to build one classification tree for each country, the decision rules of which at every splitting point is an average of the corresponding rules from other similar classification trees. The latter trees are fitted on slightly different samples. To put it more simply, the idea is to exclude some observations from the dataset, fit a classification tree on the remaining ones and repeat; finally, extract one classification tree the decision rules of which are an average of all previously fitted trees' rules. The average rule is calculated because, for interpretability reasons, the main goal is to have found *a single* rule at the end of the day (at least, one per country). The motivation for this idea was the need for this paper to apply a method which, on the one hand, is as easily interpretable as a Decision Trees model and, on the other, is not too sensitive for overfitting as a Random Forests model (ideally). The kind of decision trees proposed is expected to be less prone to overfitting than simple ones, in the sense that the former are not based on a single dataset but built through a resampling procedure. This is why they are called robust in the context of this paper.

For this purpose, two functions were written in R, which differ only in the way they select the observations excluded. The first function does this without replacement, in a way identical to the sampling procedure of *K*-fold cross-validation. The second, instead of *K*-fold sampling, uses a random sampling procedure with replacement. Algorithm 1 describes in more detail the method proposed:

Algorithm 1 The Average Trees algorithm

INPUT: dt, $imp = \{v_1, v_2, \dots\}$, nt, s // Dataset, important variables, number of trees to fit[15] and size of excluded sample per iteration, respectively. //

Step 1: For each $i = 1,2, \dots, nt$:

1) Exclude s observations from dt. // Sampling method: depending on which function has been selected. //
2) Fit a decision tree on remaining data.
3) Extract the decision rules from the model fitted and save them in a table named RS.

14. More details about how a pre-recessionary period was defined can be found in Appendix, A.3, in reference "b)" about function def.recess(dt).
15. This parameter is available only in the function which performs random sampling with replacement. In the other one, the maximum number of trees is restricted by the number of excluded observations per iteration (here, parameter *s*). Thus, in that case, *nt* is calculated by the algorithm.

Step 2: Search in *RS* for the most frequent tree structure (i.e., for the moment, ignore the numerical part of each rule and only care about the splitting variable and the inequality direction) and select it.

Step 3: Take into account only the trees that have the most frequent structure and, at each split, compute the average rule from the numerical parts previously excluded. These average rules, which are based on the most frequent tree structure, form the *average tree*.

OUTPUT: *average tree*

For example, let us assume that 90% of the trees fitted have the simple rule that, if variable $V_1 > a$, then *PreRecess* = 1; otherwise, *PreRecess* = 0, where, in general, number α differs from tree to tree. Additionally, the remaining 10% are trees that have the rule: if $V_2 > b$ then *PreRecess* = 1; otherwise *PreRecess* = 0 (again, the values of b are generally different for each tree of this structure). In this case, Algorithm 1 drops out the second tree structure, because only 10% of the trees fitted have it. Consequently, the majority rule has been applied here in order to find the most prevalent tree structure. After that, Algorithm 1 keeps only the first structure and, in place of *a*, it substitutes every *a* with the average value. This is the *average tree*. The rationale behind these steps is that, since a tree structure appears more frequently than any other in different samples of the same population, it is more likely that this is the structure that best represents reality. As these trees differ only in the numerical values of their decision rules, we take the average rule by averaging these values.

The evaluation metrics

A plethora of statistical and machine learning models were compared to each other in terms of six evaluation metrics for each country. These evaluation metrics are based on the so-called *confusion matrix* for discrete classifiers of binary classification problems. The confusion matrix has the following form (Aggarwal, 2015, pp. 637-638):

Table 3. The confusion matrix in binary classification problems.

		Predicted class	
		Class-1	Class-0
Actual Class	Class-1	True Positives (***TP***)	False Negatives (***FN***)
	Class-0	False Positives (***FP***)	True Negatives (***TN***)

At this point we form the convention that instances of class "1" are called *positives* and instances of class "0" are called *negatives*. The sum of true positives and true negatives is the number of correctly classified observations; the rest are wrongly classified. Let ***CM*** be the squared confusion matrix that contains numbers *TP*, *FN*, *FP* and *TN* from Table 3. Then, we can define the following evaluation metrics for a binary discrete classifier *c*:

$$acc(c) = \frac{TP + TN}{TP + FN + FP + TN} \tag{1}$$

$$recall(c) = \frac{TP}{TP + FN} \tag{2}$$

$$precision(c) = \frac{TP}{TP + FP} \tag{3}$$

$$specificity(c) = \frac{TN}{FP + TN} \tag{4}$$

$$falarm(c) = \frac{FP}{TN + FP} \tag{5}$$

Eq. 1 is called *classification accuracy* and it is a metric which simply denotes the overall proportion of well classified instances. Eq. 2, also known as *sensitivity* or *true positive rate* (TPR), is the proportion of instances correctly classified as 'positives' from among all truly positive instances. Eq. 3 is the proportion of instances correctly classified as 'positives' from among all instances classified as 'positives.' Eq. 4, also called *true negative rate* (TNR), is analogous to the sensitivity for negative instances, and *false alarm* of Eq. 5, also known as *false positive rate* (FPR), is the proportion of instances wrongly classified as 'positives' from among all truly negative instances.Another evaluation metric, based on *recall* and *precision*, is the F_β-score:

$$F_\beta(c) = (1 + \beta^2)\frac{recall(c) * precision(c)}{recall(c) + [\beta^2 * precision(c)]} \tag{6}$$

It may be the case that we care about both *recall* and *precision* and we want a single metric for these two. With F_β-score we take both into account and we can adjust their weights by choosing the appropriate β. A higher β means that more emphasis is placed on *recall*, while a lower β attributes more weight to *precision* (i.e. $F_\beta(c) \rightarrow recall(c)$ and $F_\beta(c) \rightarrow precision(c)$, respectively). For example, if we use classifier *c* for predicting recessions, it is not obvious – without further investigation – whether the cost of *not predicting* a recession or that of *wrongly preparing for* a recession is

greater. A policymaker may acquire this knowledge after some investigation and (s) he can adjust β accordingly, in order to incorporate this cost-centred perspective into an evaluation metric. If $\beta = 1$ we have the so-called F_1-score, which is simply the harmonic mean of *recall* and *precision*.

A short presentation of the methods used

In this subsection we briefly present the basic components of the methods used to produce this study's results. References for detailed descriptions of all of them are presented in Appendix, A.1.

- Logit: A Logit model is described by the following equation:

$$\log\left(\frac{p_i}{1-p_i}\right) = \boldsymbol{x}_i^T\boldsymbol{\beta}\,, \tag{7}$$

where p_i is the probability that the response (binary) variable equals 1, $\boldsymbol{x}_i^T$ is the 1 x (K+1) vector of i-th observation values on K independent variables plus a constant, and $\boldsymbol{\beta}$ is the (K+1) x 1 vector of K parameter values plus a constant.

- Probit: With regard to Probit models, according to Baltagi (2002, pp. 332-333), these differ from Logit only in the tails of their CDFs[16]. Both have the CDF of a t-distribution; Probit has the one with infinite degrees of freedom, while Logit has that of seven. More specifically, Probit has the following CDF:

$$p_i = \Phi(\boldsymbol{x}_i^T\boldsymbol{\beta}) = \int_{-\infty}^{\boldsymbol{x}_i^T\boldsymbol{\beta}} \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{u^2}{2}\right)du \tag{8}$$

- Support Vector Machines (SVM): In a general context, we can split a K-dimensional space into two regions using a (K-1)-dimensional *hyperplane*. The mathematical definition of a (K-1)-dimensional hyperplane is given by the following equation[17]:

$$\beta_0 + \sum_{k=1}^{K}\beta_k X_k = 0\,. \tag{9}$$

16. Cumulative Distribution Function.
17. Using words, hyperplane is a flat affine (i.e., not necessarily passing through the origin) subspace of dimension *one less* than its surrounding space (James *et al.*, 2013, p. 338).

Any point $\boldsymbol{x}$, the coordinates of which satisfy Eq. 9, lies on the hyperplane. But it may be the case that when substituting a point's coordinates in the LHS of Eq. 9, the result is either > 0 or < 0, instead of being equal to zero. This simply means that this point is either 'above' or 'below' the hyperplane in the *K*-dimensional space. This idea can be used for constructing classifiers by using proper hyperplanes as decision boundaries, which divide the *K*-dimensional space into two regions, one for each class of observations. In the context of SVM, this hyperplane is the result of the following optimisation problem:

$$\underset{\beta_0,\beta_1,\ldots,\beta_K,\varepsilon_1,\varepsilon_2,\ldots,\varepsilon_n}{\text{maximize}} \; W \tag{10}$$

$$\text{subject to} \sum_{k=1}^{K} \beta_k^2 = 1, \tag{11}$$

$$y_i(\beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \cdots + \beta_K X_{iK}) \geq W(1 - \varepsilon_i), \tag{12}$$

$$\varepsilon_i \geq 0, \sum_{i=1}^{n} \varepsilon_i \leq C, \tag{13}$$

where W is the width of the margin[18], β_ks are the parameters which define the hyperplane along with X_ks, ε_is, which are slack variables[19], each $y_i \in \{-1,1\}$represents the class of the i-th observation, and C is a tuning parameter for the tolerance of margin violations. If $C = 0$ no margin violations are allowed. It can be shown that the solution of the above optimisation problem involves only the inner products[20] of observations, which leads to the conclusion that the linear classifier can be represented as:

$$f(\boldsymbol{x}) = \beta_0 + \sum_{i=1}^{n} \alpha_i \langle \boldsymbol{x}, \boldsymbol{x}_i \rangle, \tag{14}$$

18. In this context, margin is the distance between a hyperplane and the closest points (i.e., observations) on either side. However, for SVM to also be used in cases where two classes cannot be perfectly separated by a hyperplane, a margin can be violated to some extent by correctly classified observations and/or by misclassifications.
19. If $\varepsilon_i > 0$, this means that the i-th observation has violated the margin and if $\varepsilon_i > 1$, this means that the i-th observation is on the wrong side of the hyperplane.
20. For two observations $\boldsymbol{x}_1, \boldsymbol{x}_2$, their inner product is defined as: $\langle \boldsymbol{x}_1, \boldsymbol{x}_2 \rangle = \sum_{k=1}^{K} x_{1k} x_{2k}$.

where α_is are parameters to be estimated[21]. Equivalently, one can write Eq. 14 as:

$$f(\boldsymbol{x}) = \beta_0 + \sum_{i=1}^{n} \alpha_i K(\boldsymbol{x}, \boldsymbol{x}_i), \tag{15}$$

where function $K(\boldsymbol{x}, \boldsymbol{x}_i)$ is called *linear kernel* and it simply computes the inner product of two vectors. The advantage of this approach is that one may very well use another kernel function in Eq. 15 – probably a nonlinear one – to produce a much more flexible decision boundary. Such an example is the *radial kernel*:

$$K(\boldsymbol{x}_i, \boldsymbol{x}_{i'}) = \exp\left(-\gamma \sum_{k=1}^{K} (x_{ik} - x_{i'k})^2\right) \tag{16}$$

The use of a radial kernel in SVM produces nonlinear margins and decision boundaries and, more specifically, boundaries of 'circular shape'. However, we should not overlook the fact that SVM is always a *linear* approach. Regardless of the kernel used, SVM's solution is always a linear decision boundary at a higher-dimensional space (larger than K). In the original feature space though, it turns out that the decision boundary is generally nonlinear.

- The k-Nearest Neighbours (k-NN) algorithm:

Algorithm 2 k-nearest neighbours (k-NN)

INPUT: $\boldsymbol{X}_{n\times J}$, $\boldsymbol{x}_{new} \in \mathbb{R}^J$, $C = \{c_1, \ldots, c_M\}$, $\boldsymbol{y} \in C^n$, $k \in \mathbb{N}^*$

/// A data matrix of n observations and J predictors, the observation to be classified, a set of M classes, the class vector and the number of k nearest neighbours, respectively. ///

1: **for** each $\boldsymbol{x}_i$ in $\boldsymbol{X}_{n\times J}$ // i from 1 to n.
2: $d_i \leftarrow Distance(\boldsymbol{x}_i, \boldsymbol{x}_{new})$
3: $D = \{(d_1, 1), \ldots, (d_n, n)\}$ // Set of 'distance-index' tuples.
4: $D_s \leftarrow Sort(D, 1)$ // Sort by ascending order, by distance.
5: $I \leftarrow Untuple(D_s, 2)$ // Extract indices.
6: $c\_nn = \{y_{I[1]}, \ldots, y_{I[k]}\}$ // Classes of k-nearest neighbours.
7: $y_{new} \leftarrow Mode(c\_nn)$ // Most frequent class of k-NN.

OUTPUT: y_{new}

21. In this formulation, parameters α_i have substituted the original β_k, $k > 0$.

By looking at Algorithm 2, one realises that the class of a new observation is determined *only* by the classes of its k-nearest neighbours. Using the majority rule, $\boldsymbol{x}_{new}$ is assigned to the class which predominates. Function *Distance* in line 2 computes the distance between two vectors and it may use any distance metric, depending on their domain. For example, such a metric is the *Euclidean distance*[22]. The underlying assumption for the k-NN classifier is that observations of the same class are similar to each other, which means that – given a distance metric – they are close to each other. This simple assumption provides the classifier with great flexibility: unlike the previously presented methods, the true shapes of decision boundaries do *not* need to be taken into consideration before solving the classification problem. In fact, decision boundaries may be of any shape and the classifier can still be reliable if number k is properly chosen. Moreover, no distributional assumptions are made.

- Decision Trees: Decision Trees is a method based on a procedure called *recursive binary splitting*:

Algorithm 3 Recursive Binary Splitting

INPUT: $\boldsymbol{X}_{n\times J}$, $\boldsymbol{x}_i \in \mathbb{R}^J$, n_{min}, $C = \{c_1, \dots, c_M\}$, $\boldsymbol{y} \in \mathbb{R}^n$ *or* $\boldsymbol{y} \in C^n$

// domain of $\boldsymbol{y}$ depends on the problem (regression or classification, respectively). //

Step 1: Choose splitting variable j and splitting point s so that quantity 17 (regression problem) or 18 (classification problem) is *minimised*:

$$\sum_{i:\, \boldsymbol{x}_i \in R_1(j,s)} \left(y_i - \hat{y}_{R_1}\right)^2 + \sum_{i:\, \boldsymbol{x}_i \in R_2(j,s)} \left(y_i - \hat{y}_{R_2}\right)^2 \tag{17}$$

$$\sum_{c \in C} \hat{p}_{R_1,c}\left(1 - \hat{p}_{R_1,c}\right) + \sum_{c \in C} \hat{p}_{R_2,c}\left(1 - \hat{p}_{R_2,c}\right) \tag{18}$$

Step 2: For each of the resulting regions repeat Step 1, until no terminal node (i.e., tree leaf) R_g contains more than n_{min} observation(s).

OUTPUT: $T_0 = \{R_1, R_2, \dots, R_G\}$ // Tree consisted of G regions.

22. It is defined as follows: $d_E(\boldsymbol{x}, \boldsymbol{y}) = \sqrt{\sum_{k=1}^{K}(x_k - y_k)^2}$ (Kubat, 2017, p. 46).

The procedure above shows how a decision tree is built. In regression problems, Algorithm 3 chooses the split (i.e., variable j and point s) which minimises the RSS of the two resulting regions $R_1(j,s) = \{X|X_j < s\}$ and $R_2(j,s) = \{X|X_j \geq s\}$. Here, $\hat{y}_{R_g} = \frac{1}{n_g}\sum_{i:\, x_i \in R_g} y_i$, where n_g is the number of observations which lie in R_g. In classification problems, Algorithm 3 minimises the *Gini index*[23] instead of RSS, as the latter cannot be used in a classification setting.
Here, $\hat{p}_{R_g,c} = \frac{1}{n_g}\sum_{i:\, x_i \in R_g}\{1_{y_i=c}\}$, where notation $1_{yi=c}$ means that this quantity equals one, if $y_i = c$ or zero otherwise. Quantity $\hat{p}_{R_g,c}$ is also an estimate of the probability that an observation of class c lies in region R_g. As recursive binary splitting algorithm builds a large tree, which is very well fitted to the training data, it becomes necessary to properly 'prune' it in order to achieve better generalisability. By and large, this is what the next steps of Decision Trees algorithm do in order to produce more accurate predictions.

- Random Forests: Random Forests is an algorithm which is based on decision tree estimators and it aims to reduce their high variance. Its general form is the following:

Algorithm 4 Random Forests

INPUT: $\boldsymbol{X}_{n\times J}$, $\boldsymbol{x}_i \in \mathbb{R}^J$, $C = \{c_1, \ldots, c_M\}$, $\boldsymbol{y} \in \mathbb{R}^n$ *or* $\boldsymbol{y} \in C^n$, n_{min}, B, n_b, h

Step 1: For $b = 1,2, \ldots, B$: Draw with replacement a random subsample[24] of size n_b from observations given.

Step 2: For each subsample build a decision tree T_b as follows:

While there is a region containing more than n_{min} observations:

1) Choose randomly h out of the J variables.
2) From these h variables, select the variable j and the point s which minimise quantity 17 or 18, depending on the problem.
3) Split variable j at point s.

OUTPUT: $RF = \{T_1, \ldots, T_B\}$

23. In general, the Gini index is defined as follows: $G = \sum_{c\in C}\hat{p}_{gc}(1-\hat{p}_{gc})$, where $\hat{p}_{gc}$ is the proportion of observations from class c in region g.
24. A so-called *bootstrap sample*.

We observe that the output of Algorithm 4 is a set of B decision trees. In order to make a single prediction, we need to perform an action called *bootstrap aggregation* – also known as *bagging*:

$$\hat{y} = \frac{1}{B}\sum_{b=1}^{B} T_b(\boldsymbol{x}) \tag{19}$$

$$\hat{y} = mode(\{T_1(\boldsymbol{x}), \dots, T_B(\boldsymbol{x})\}), \tag{20}$$

where Eq. 19 is used for predictions in regression problems and Eq. 20 for predictions in classification problems. Notation $T_b(\boldsymbol{x})$ indicates the prediction of tree T_b, given observation $\boldsymbol{x}$.

- Boosted Regression Trees (BRT): BRT algorithm is an extension of the Decision Trees algorithm and, specifically, of regression trees. It has the following form:

Algorithm 5 Boosted Regression Trees (BRT)

INPUT: $\boldsymbol{X}_{n\times J}$, $\boldsymbol{x}_i \in \mathbb{R}^J$, $\boldsymbol{y} \in \mathbb{R}^n$, B, λ, d // Data and tuning parameters.

Step 1: Set $\hat{f}(\boldsymbol{x}) = 0$ and $r_i = y_i \; \forall \; i \in \{1, \dots, n\}$, where r_i represents the residual of the i-th observation. // $\boldsymbol{r}$ is a vector of residuals.

Step 2: For $b = 1,2, \dots, B$ repeat:

a) Fit a tree $\hat{f}^b$ with d splits (i.e., $d+1$ terminal nodes) to the training data $(\boldsymbol{X}, \boldsymbol{r})$.

b) Update $\hat{f}$ by adding in a shrunken version of the new tree:

$$\hat{f}(\boldsymbol{x}) \leftarrow \hat{f}(\boldsymbol{x}) + \lambda \hat{f}^b(\boldsymbol{x}) \tag{21}$$

c) Update residuals:

$$r_i \leftarrow r_i - \lambda \hat{f}^b(\boldsymbol{x}_i) \tag{22}$$

Step 3: The boosted model $\hat{f}(\boldsymbol{x})$ is:

$$\hat{f}(\boldsymbol{x}) = \sum_{b=1}^{B} \lambda \, \hat{f}^b(\boldsymbol{x}) \tag{23}$$

OUTPUT: $\hat{f}(\boldsymbol{x})$

Boosting is a technique that builds an estimator sequentially (James *et al.*, 2013, p. 321). This means that, at each step, information from the previously constructed estimator(s) is used, which is what Step 2 of Algorithm 5 does.

- Artificial Neural Networks (ANN): We focus on the simplest class of ANN, the *feed-forward neural network*, also known as *multilayer perceptron* (MLP). We still use notation $X_1,...,X_K$ for independent (explanatory) variables and Y for the dependent variable. In the ANN context, we call the former *input variables*, because they are the input of an ANN system. Similarly, the result related to Y (either a real value or a class probability, for regression or classification problems, respectively) is the *output* of an ANN system. What we want to do with an ANN model is to approximate a nonlinear function f, for which:

$$y(\boldsymbol{x}, \boldsymbol{w}) = f\left(\sum_{j=1}^{M} w_j \varphi_j(\boldsymbol{x})\right), \tag{24}$$

where y corresponds to the output, $\boldsymbol{x}$ is an observation vector of length K, $\boldsymbol{w}$ is a vector of some weight parameters, f is called *activation function*[25] and φ is a nonlinear *basis function* (i.e., a nonlinear transformation of vector $\boldsymbol{x}$). We observe that output y is a nonlinear transformation of M linear combinations. The following equations compose the mathematical representation of a feed-forward neural network model:

$$a_j = \sum_{k=1}^{K} w_{jk}^{(1)} x_k + w_{j0}^{(1)}, \qquad j \in \{1, \dots, M\}. \tag{25}$$

Parameters w_{jk} are called *weights*, parameters w_{j0} are the *biases*[26], a_js are called *activations* and superscript '(1)' denotes that the corresponding quantities belong to the first layer of the network. For each a_j there is a differentiable nonlinear activation function h:

$$z_j = h(a_j) \tag{26}$$

Quantities z_j are called *hidden units*. In general, h is chosen to be a sigmoidal function. The following holds for the second layer:

$$a_l = \sum_{j=1}^{M} w_{lj}^{(2)} z_j + w_{l0}^{(2)}, \qquad l \in \{1, \dots, L\}. \tag{27}$$

25. Which means that, given some input, it is a function that determines the output.
26. Here, the term *bias* is used to describe a parameter w_0 that allows any fixed offset in the data (Bishop, 2006, p. 138). It is often useful to define $\varphi_0(\boldsymbol{x}) = 1$. In the linear regression context, the bias parameter is the intercept.

If there are only two layers, then L is the number of outputs and quantities a_l are called *output unit activations*. Finally, the output units are calculated as follows:

$$y_l = \sigma(a_l), \tag{28}$$

where activation function σ is the identity function for regression problems, the logistic sigmoid function for (multiple) binary classification problems (Eq. 29) or the softmax function (Bishop, 2006, p. 198) for multiclass classification problems (Eq. 30):

$$\sigma(a_l) = \frac{1}{1 + \exp(-a_l)} \tag{29}$$

$$\sigma(\alpha_l) = \frac{\exp(a_l)}{\sum_{r=1}^{L} \exp(a_r)} \, . \tag{30}$$

Eqs. 29 and 30 represent the conditional probability for an observation to belong to class l given $\boldsymbol{x}$, i.e. they give values in [0,1]. Putting it all together, the MLP model takes the following functional form:

$$y_l(\boldsymbol{x}, \boldsymbol{w}) = \sigma\left(\sum_{j=1}^{M} w_{lj}^{(2)} h\left(\sum_{k=1}^{K} w_{jk}^{(1)} x_k + w_{j0}^{(1)}\right) + w_{l0}^{(2)}\right), \tag{31}$$

$$\forall \, l \in \{1, \ldots, L\} \, .$$

For more than two layers, the generalisation is straightforward.

Results

In the previous section, we presented the important methodological aspects of this paper. In this section, we focus on result evaluation. The first question that needs to be answered is whether the Average Trees algorithm can provide more reliable results than classic Decision Trees and Random Forests. Subsequently, another question is about which method(s) perform(s) better in our datasets. And, apart from the questions related to methods, we need to provide an answer to the question about which economic theory seems more plausible according to the evidence – if any of them stands out. The results presented in this section contribute towards finding answers to these questions.

The evaluation procedure was based on K-fold cross-validation. Parameter K was chosen for each country to ensure a sufficient number of observations is included in test sets[27] but, in parallel, that the training sets also are of a sufficient size for building

27. In general, the target was fifteen observations.

the most reliable models possible. The following metrics were used for evaluating each method: 1) Classification Accuracy, 2) Sensitivity, 3) Precision, 4) Specificity, 5) False Alarm and 6) F_1-Score. For each method, the results from the K-fold cross-validation are summarized by calculating a *weighted average* on each metric. The weights are proportional to test set sizes. It is expected that the K-th test set is usually of smaller size than the previous K-1 test sets, because the division of observation number by K is likely to give a non-zero remainder. The last test set may show, for example, 100% Classification Accuracy, since it consists of only one (correctly classified) observation. But it is obvious that such results do not have the same significance as those from the other K-1 test sets. Thus, the impact of the last test set should be shrunk proportionally to the number of cases used for evaluating a classifier on this set. This detail is taken care of by using the weighted average mentioned above.

The ten methods evaluated are the following[28]: Average Trees (both variants of the algorithm; i.e., K-fold sampling and random sampling with replacement), Decision Trees, Random Forests, Logit, Probit, k-NN, Boosted Regression Trees (Logistic Regression model), Support Vector Machines, and Artificial Neural Networks (single-layer, feed-forward).

It has been mentioned that the important variables were selected through a procedure based on Random Forests. This was a first step to reasonably reduce the number of variables from ~150 to ~10, according to their MDG. However, depending on the method, an additional approach was followed in order to further decrease model complexity, in cases where such a decrease improves the generalisability of a model. To be more specific, for the Logit and Probit models a stepwise forward selection was applied, using AIC as the model selection criterion. For the k-NN algorithm, Principal Component Analysis (PCA)[29] was applied before feeding the model with data, in order to avoid the 'curse of dimensionality'[30]. The number of principal components selected was defined manually for each country, by looking at the output of R function plot.prcomp(), which shows the proportion of variance explained by each principal component. The rationale behind the choice of this number is subjective and similar to that of the initial variable selection based on MDG. As for recursive partitioning methods (including the Average Trees), their model complexity was left

28. The corresponding R packages used are the following (explanation is given if package and function have different names): rpart, randomForest, stats (for glm()), class (for knn()), gbm, e1071 (for svm()) and nnet.
29. More details about this method can be found in James *et al.* (2013, pp. 230-237).
30. According to Kubat (2017, p. 54), this term describes a situation where, as the number of explanatory variables increases, it becomes less likely that two observations are close to each other in the high-dimensional space. Therefore, it is hard to distinguish whether the large distance between them indicates class differentiation or not. Nevertheless, one may overcome this problem by increasing the number of observations or by applying a dimension reduction technique, such as the PCA.

to be controlled by the internal R procedures; i.e., the relevant parameters remained at their default values. For the BRT, parameter shrinkage[31] was chosen such that the out-of-sample Classification Accuracy would be the largest. Regarding the SVM, a range of values were tried for each country regarding parameters C and γ[32]. A three-dimensional plot was proved very helpful for finding 'parameter areas' of high Classification Accuracy (an example is presented in Appendix, A.5). The values selected were those that gave the largest mean out-of-sample Classification Accuracy. Finally, regarding the ANN, a *weight decay regularization*[33] was applied by searching over a set of values that provided better out-of-sample performance. Note that for any method used there is no guarantee for a globally optimal parameter setting, since the latter was selected by trial and error.

Australia

The presentation of evaluation results begins with Australia. In this dataset, data are from 1972.Q1 to 2014.Q4 (length: 43 years – 172 observations). Figure 1 shows the results of varImpPlot() for the initial variable selection.

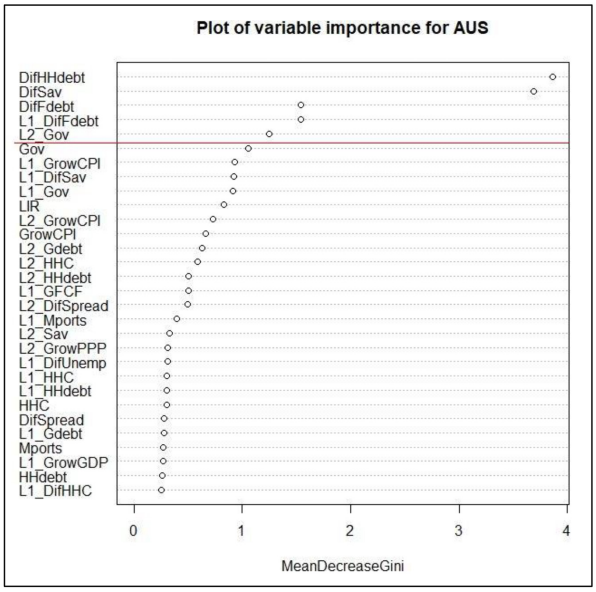

Figure 1. The five – out of 131 – variables chosen for Australia (those above the red line).

31. This is parameter λ from Eq. 23.
32. See also optimization problem 10-13 and Eq. 16. Apart from the radial kernel, the sigmoid was also tried. It is defined as: $K(\boldsymbol{x}_i, \boldsymbol{x}_{i'}) = \tanh(\gamma \boldsymbol{x}_i^T \boldsymbol{x}_{i'} + r)$. For reasons related to computation time, parameter r was left at default value 0.
33. Weight decay regularization is a method for reducing potential overfitting of an ANN model. This is done by decaying some weight parameters towards zero. More details can be found in Bishop (2006, pp. 256-257).

Table 4 contains the out-of-sample performance of the methods described above in six evaluation metrics. In Appendix, A.8, one can also find the corresponding tables for the in-sample performance of the methods presented. Their parameterisation is the same as that emerged from the out-of-sample evaluation procedure. Moreover, test sets are also of the same size; the difference is that the training sets consist of all available observations. Lastly, in Figure 8 one can visually compare the performance of all methods presented.

There was some doubt about including *L2_Gov* in the set of important variables. As their total number is relatively small, the final decision was to include it.

Table 4. Evaluation results – Australia. Each cell is the average of the results of each test set. Regarding the *K*-fold cross-validation procedure: $K = 12 \rightarrow$ test set size=15.

Method	Classification Accuracy	Sensitivity	Precision	Specificity	False Alarm	F_1-Score	Comments
Average Trees (*K*-fold)	81.98%	45%	20.54%	88.59%	11.41%	37.96%	ex.size = 0.12
Average Trees (random)	86.63%	53.57%	42.71%	90.92%	9.08%	54.86%	ex.size = 0.45, tree.nr = 200
Decision Trees	81.98%	45%	20.54%	88.59%	11.41%	37.96%	probability threshold: 0.5
Random Forests	88.37%	55%	47.78%	94.39%	5.61%	61.11%	probability threshold: 0.6
Logit	88.95%	45%	43.33%	96.28%	3.72%	53.33%	probability threshold: 0.75
Probit	89.53%	35%	54.17%	97.45%	2.55%	45%	probability threshold: 0.75
k-NN	90.7%	60%	68.89%	96.4%	3.6%	**79.05%**	k = 6, # of principal components: 2
BRT	90.7%	45%	**77.78%**	**98.41%**	**1.59%**	76.19%	shrinkage = 0.013, probability threshold: 0.45
SVM	**92.44%**	86.43%	63%	92.76%	7.24%	70.74%	γ=0.0005, C=1.3, sigmoid kernel, probability threshold: 0.15
ANN	90.12%	**88.57%**	52.06%	89.92%	10.08%	69.43%	probability threshold: 0.54, # of hidden nodes: 5

The probability thresholds mentioned in some cells of "Comments" column refer to the *least* estimated probability for which the predicted class is "1 – Recession[34]"; i.e., any estimated probability under the threshold gives the prediction "0 – No Recession". These thresholds were chosen so that the value of Classification Accuracy is the maximum possible[35]. Methods with no reference to probability thresholds are those

34. Actually, class "1" refers to a pre-recessionary period.
35. Thresholds that produced Sensitivity=0 or False Alarm=1 were not considered in any dataset.

that do not provide predictions in probabilistic form. Regarding the parameters of Average Trees algorithm, ex.size refers to the percentage of observations to be excluded[36] and tree.nr refers to parameter *nt* from Algorithm 1.

Germany

We move on to the German dataset. The data in this case are from 1973.Q1 to 2014. Q4 (length: 42 years – 168 observations).

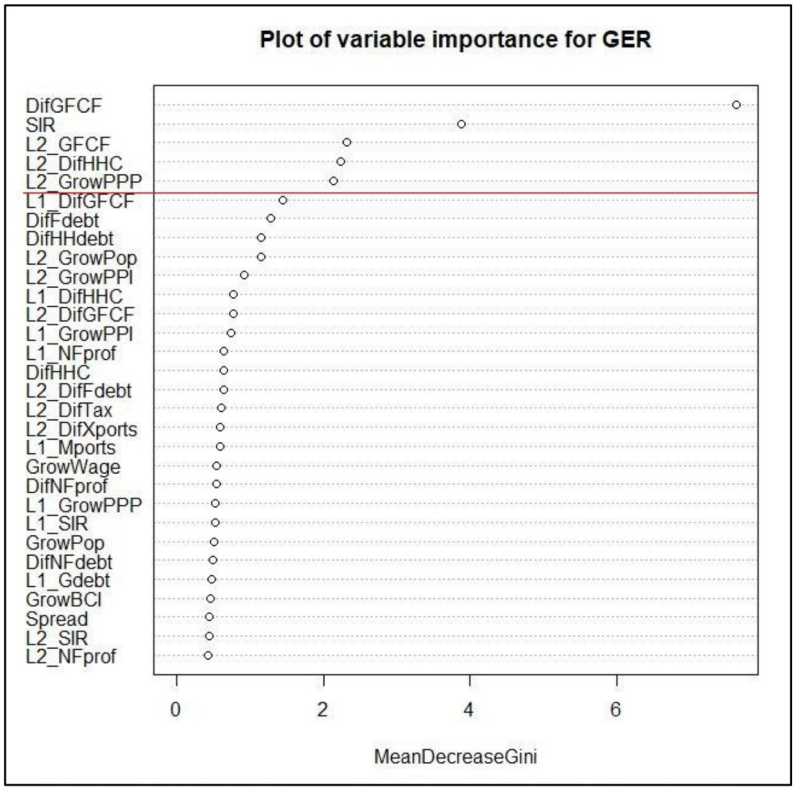

Figure 2. The five – out of 146 – variables chosen for Germany.

36. In Algorithm 1, this quantity is expressed in integer form (parameter *s*).

Table 5. Evaluation results – Germany[37].
Regarding the *K*-fold cross-validation procedure: K=11 →test set size=16.

Method	Classification Accuracy	Sensitivity	Precision	Specificity	False Alarm	F_1-Score	Comments
Average Trees (*K*-fold)	82.74%	43.75%	67.86%	94.44%	5.56%	59.34%	ex.size = 0.13
Average Trees (random)	84.52%	50%	72.22%	95.24%	4.76%	65.56%	ex.size=0.225, tree.nr=200
Decision Trees	80.95%	18.75%	75%	98.41%	1.59%	50%	probability threshold: 0.8
Random Forests	86.9%	**71.88%**	71.88%	91.27%	8.73%	68.54%	probability threshold: 0.37
Logit	83.93%	31.25%	**88.89%**	**99.21%**	**0.79%**	**85.71%**	probability threshold: 0.85
Probit	83.93%	31.25%	**88.89%**	**99.21%**	**0.79%**	**85.71%**	probability threshold: 0.85
k-NN	**87.5%**	53.13%	71.43%	95.63%	4.37%	72.38%	k = 3, # of principal components: 3
BRT	85.71%	68.75%	72.08%	90.48%	9.52%	66.96%	shrinkage = 0.01, probability threshold: 0.3
SVM	85.12%	59.38%	57.21%	90.48%	9.52%	60.93%	γ=0.0014, C=2.9, radial kernel, probability threshold: 0.35
ANN	79.76%	54.69%	52.5%	84.52%	15.48%	55.26%	probability threshold: 0.67, # of hidden nodes: 5

37. Regarding the number of principal components in the case of *k*-NN algorithm, the initial choice based on the visual output of plot.prcomp() was to choose two principal components. However, it was discovered that choosing three improves mean Classification Accuracy.

Japan

We proceed to the dataset of Japan. In this case, the data are from 1973.Q1 to 2015. Q4 (length: 43 years – 172 observations).

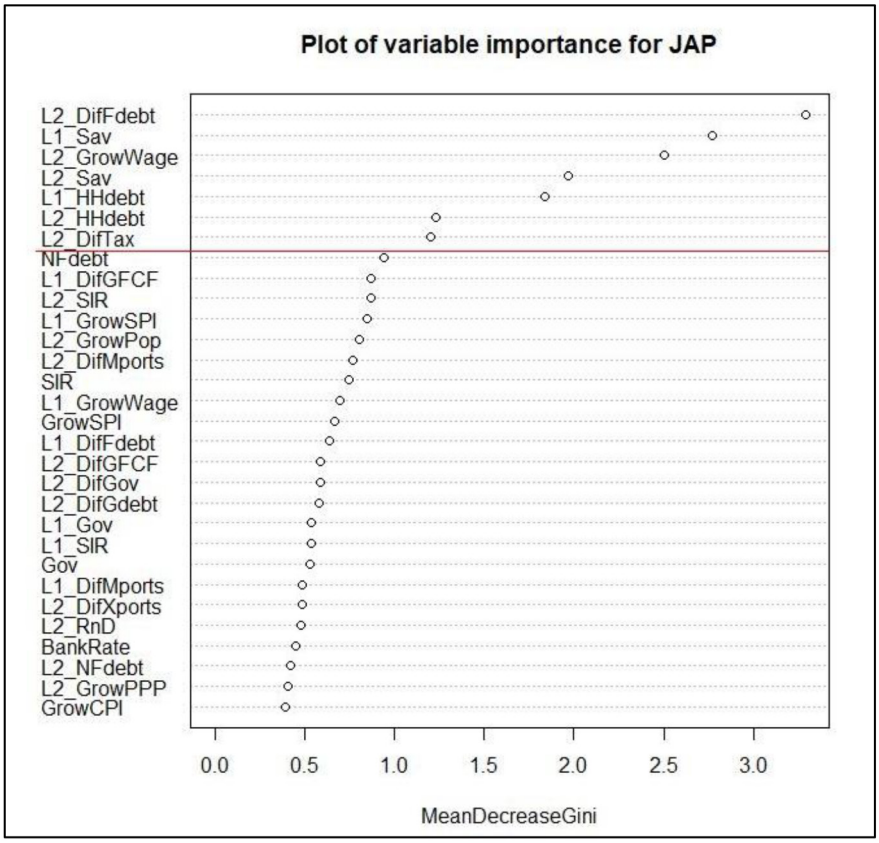

Figure 3. The seven – out of 128 – variables chosen for Japan.

Table 6. Evaluation results – Japan[38].
Regarding the *K*-fold cross-validation procedure: K=12 →test set size=15.

Method	Classification Accuracy	Sensitivity	Precision	Specificity	False Alarm	F_1-Score	Comments
Average Trees (*K*-fold)	81.4%	32.93%	38.16%	90.63%	9.37%	48.96%	ex.size = 0.09
Average Trees (random)	81.4%	32.93%	38.16%	90.63%	9.37%	48.96%	ex.size = 0.1, tree.nr = 200
Decision Trees	83.14%	8.54%	33.33%	**98.64%**	**1.36%**	50%	probability threshold: 0.8
Random Forests	86.63%	24.8%	**77.27%**	94.19%	5.81%	**83.77%**	probability threshold: 0.45
Logit	83.72%	42.07%	31.98%	86.38%	13.62%	49.66%	probability threshold: 0.35
Probit	84.3%	42.07%	31.98%	86.96%	13.04%	49.66%	probability threshold: 0.35
k-NN	86.05%	39.02%	61.19%	95.2%	4.8%	60.9%	k = 2, # of principal components: 3
BRT	81.98%	13.11%	27.65%	96.26%	3.74%	32.95%	shrinkage = 0.035, probability threshold: 0.3
SVM	**89.53%**	**81.71%**	61.35%	87.1%	12.9%	75%	γ=0.0049, C=1.2, radial kernel, probability threshold: 0.3
ANN	70.93%	63.41%	33.41%	66.33%	33.67%	57.87%	probability threshold: 0.5, # of hidden nodes: 6

38. Similarly, in this dataset the decision based on the output of plot.prcomp() was to select the first two principal components, but the addition of a third one improved Classification Accuracy.

Mexico

The next country to be presented is Mexico. Data in the Mexican dataset are from 1973.Q2 to 2015.Q4 (length: 42.75 years – 171 observations). In Figure 4 we see, for the first time so far, two trade-related variables at the top of the table.

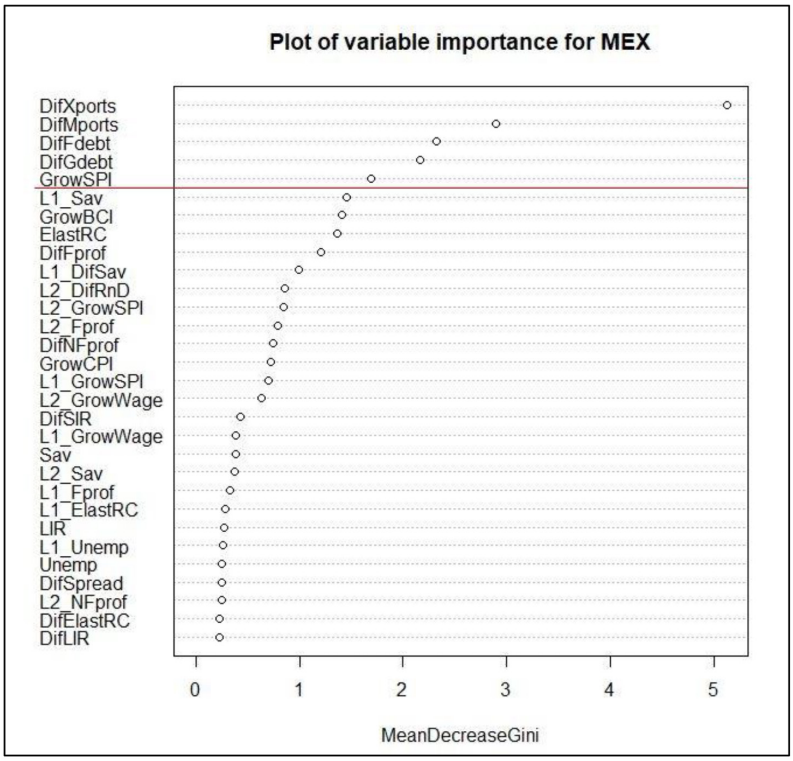

Figure 4. The five – out of 128 – variables chosen for Mexico.

The following Table 7 is the first table that contains results from an ensemble model. As no single model provided satisfactory results for the case of Mexico, an ensemble model was built in order to combine the strengths of the best three models into one and, hopefully, mitigate their weaknesses[39]. It makes a prediction by applying the majority rule on the predictions of the three models selected. Lastly, it may seem somewhat strange that the Average Trees algorithm with random sampling has mean F_1-Score 73.33%, while the corresponding values of Sensitivity and Precision are 30% and 41.67%, respectively. Such peculiarities may be found in other datasets, too, since the F_1-Score is calculated *only if* both Sensitivity and Precision values exist. Particularly for this dataset, only two values were calculated for the F_1-Score of Average Trees (random), because both Sensitivity and Precision were simultaneously real numbers for only two test sets.

39. Explanations about this topic are presented in Discussion.

Table 7. Evaluation results – Mexico.
Regarding the *K*-fold cross-validation procedure: K=12 → test set size=15.

Method	Classification Accuracy	Sensitivity	Precision	Specificity	False Alarm	F_1-Score	Comments
Average Trees (*K*-fold)	81.29%	**50%**	35.56%	87.77%	12.23%	66.23%	ex.size = 0.12
Average Trees (random)	82.46%	30%	41.67%	92.56%	7.44%	**73.33%**	ex.size = 0.1, tree.nr = 200
Decision Trees	80.7%	**50%**	34.67%	86.98%	13.02%	64.76%	probability threshold: 0.5
Random Forests	84.21%	25%	50%	95.11%	4.89%	46.67%	probability threshold: 0.6
Logit	84.21%	5%	20%	96.81%	3.19%	22.22%	probability threshold: 0.5
Probit	81.29%	5%	5.56%	93.04%	6.96%	10%	probability threshold: 0.4
k-NN	82.46%	22.5%	25.71%	92.72%	7.28%	27.78%	k = 1, # of principal components: 4
BRT	**85.96%**	20%	43.75%	97.24%	2.76%	38.33%	shrinkage = 0.01 probability threshold: 0.3
SVM	84.8%	22.5%	**68.18%**	94.42%	5.58%	37.78%	γ=0.006, C=0.08, radial kernel, probability threshold: 0.16
ANN	85.38%	42.5%	48.81%	92.66%	7.34%	46.21%	probability threshold: 0.55, # of hidden nodes: 3
Ensemble	**85.96%**	20%	50%	**97.45%**	**2.55%**	41.9%	Majority rule: {Average Trees (*K*-fold), BRT, ANN}

United Kingdom (UK)

We move on to the dataset for the United Kingdom. In this case, the data are from 1973.Q2 to 2014.Q4 (length: 41.75 years – 167 observations).

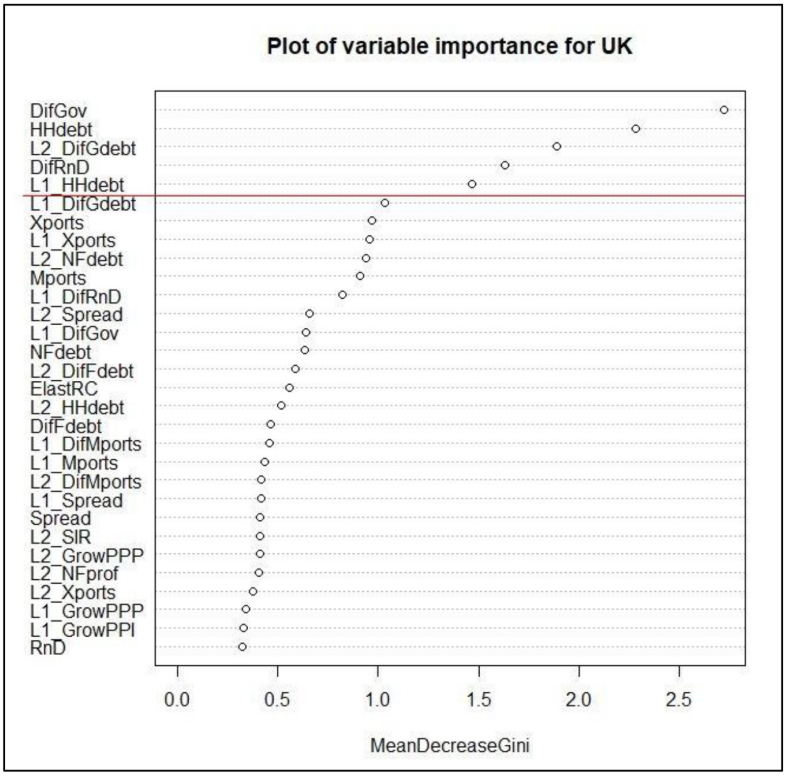

Figure 5. The five – out of 149 – variables chosen for the UK.

Table 8. Evaluation results – United Kingdom.
Regarding the *K*-fold cross-validation procedure: K=11 →test set size=16.

Method	Classification Accuracy	Sensitivity	Precision	Specificity	False Alarm	F_1-Score	Comments
Average Trees (*K*-fold)	88.62%	**50%**	41.27%	94.13%	5.87%	76.36%	ex.size = 0.25
Average Trees (random)	88.62%	**50%**	41.27%	94.13%	5.87%	76.36%	ex.size = 0.15, tree.nr = 200
Decision Trees	88.62%	43.75%	35.42%	95.13%	4.87%	77.5%	probability threshold: 0.5
Random Forests	88.62%	33.33%	**64.58%**	96.33%	3.67%	64.29%	probability threshold: 0.7
Logit	88.02%	20.83%	62.5%	97.13%	2.87%	71.43%	probability threshold: 0.5
Probit	88.02%	20.83%	62.5%	97.13%	2.87%	71.43%	probability threshold: 0.5
k-NN	**89.82%**	37.5%	60%	**98.6%**	**1.4%**	**77.78%**	k = 10, # of principal components: 2
BRT	89.22%	43.75%	63.33%	95.53%	4.47%	73.33%	shrinkage = 0.01 probability threshold: 0.3
SVM	85.63%	12.5%	22.22%	93.41%	6.59%	47.06%	γ=0.0003, C=0.009, radial kernel, probability threshold: 0.95
ANN	86.83%	12.5%	22.22%	96.41%	3.59%	57.14%	probability threshold: 0.51, # of hidden nodes: 4
Ensemble	88.62%	**50%**	41.27%	94.13%	5.87%	76.36%	Majority rule: {Average Trees (random), BRT, *k*-NN}

United States of America (USA)

The dataset for the USA consists of data from 1973.Q1 to 2014.Q4 (length: 42 years – 168 observations).

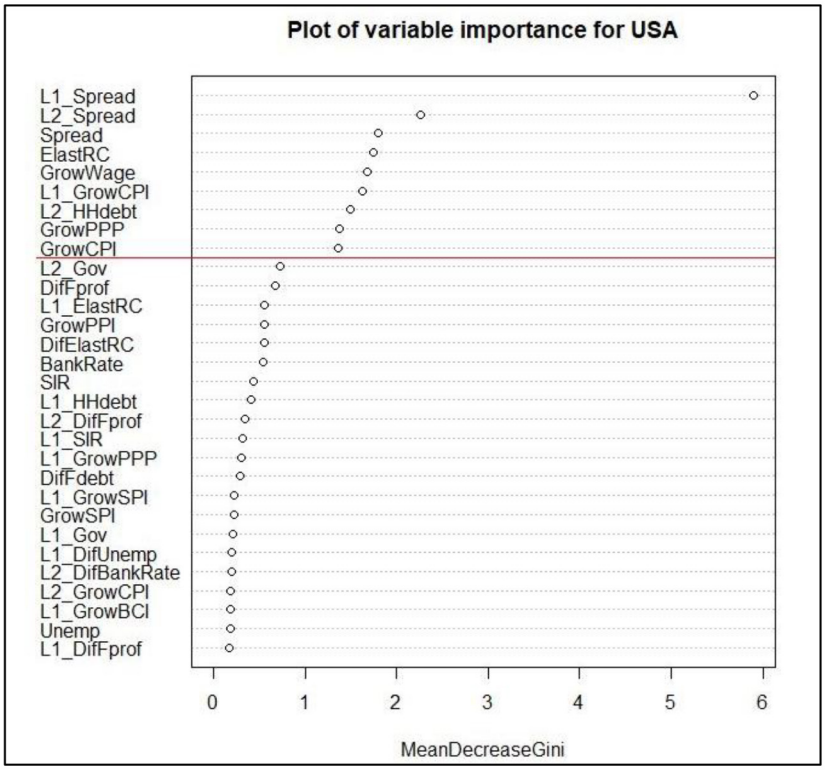

Figure 6. The nine – out of 149 – variables chosen for the USA.

Table 9. Evaluation results – USA.
Regarding the *K*-fold cross-validation procedure: K=11 →test set size=16.

Method	Classification Accuracy	Sensitivity	Precision	Specificity	False Alarm	F_1-Score	Comments
Average Trees (*K*-fold)	86.9%	50%	46.94%	90.48%	9.52%	50.38%	ex.size = 0.1
Average Trees (random)	87.5%	55%	48.61%	90.48%	9.52%	53.72%	ex.size = 0.1, tree nr = 200
Decision Trees	86.9%	50%	46.94%	90.48%	9.52%	50.38%	probability threshold: 0.5
Random Forests	90.48%	50%	63.49%	95.24%	4.76%	69.96%	probability threshold: 0.4
Logit	91.07%	45%	65.56%	96.83%	3.17%	67.41%	probability threshold: 0.8
Probit	92.26%	45%	76.67%	**98.41%**	**1.59%**	74.07%	probability threshold: 0.92
k-NN	92.26%	55%	**77.78%**	96.83%	3.17%	81.9%	k = 10, # of principal components: 2
BRT	91.07%	40%	73.33%	97.62%	2.38%	**84.44%**	shrinkage = 0.009, probability threshold: 0.5
SVM	**93.45%**	**75%**	73.33%	95.24%	4.76%	71.43%	γ=0.00025, C=0.92, radial kernel, probability threshold: 0.25
ANN	88.69%	65%	42.98%	90.67%	9.33%	62.93%	probability threshold: 0.52, # of hidden nodes: 5

Discussion

In order to specify the conditions that precede economic recessions, the author's choice was to develop a method based on Decision Trees. The goal of the so-called 'Average Trees algorithm' is to provide results with the straightforward interpretability of the classic Decision Trees and the robustness of Random Forests. The choice of developing a method based on Decision Trees was made because our central topic is to specify the macroeconomic conditions before a recession commences. Decision trees provide results in a form that best suits this purpose. The question is whether Average Trees have better out-of-sample performance than classic Decision Trees and/or Random Forests.

Before answering this question, one can realise from Tables 4 – 9 that the *K*-fold variant of Average Trees *never* provided better mean Classification Accuracy than the variant of random sampling with replacement. Therefore, we can argue that random sampling improves the generalisability of the method, probably because, at each iteration, the observations excluded are not from a very specific period of the dataset, but they may be from *any* period with the same probability. So, for this vari-

ant, the information that is considered during the training phase is generally from the entire available time span. Thus, the main question is whether the performance of the random sampling variant of Average Trees exceeds that of Decision Trees and/or Random Forests. With the single exception of the Japanese dataset, classic Decision Trees never showed better performance than the Average Trees random sampling variant in terms of mean Classification Accuracy. Indeed, in four out of the remaining five datasets, Decision Trees performance was worse than that of Average Trees (random sampling). However, in no dataset did the random sampling variant of Average Trees have better performance than Random Forests. Thus, after this first application of the Average Trees algorithm on real datasets, we can say that its models tend to generalise better than classic Decision Trees without losing their straightforward interpretability. However, it seems that Average Trees cannot achieve better performance than Random Forests.

The Average Trees algorithm was developed for this paper in order to identify rules that lead to recessions. However, before looking at any data, it was not known whether such a concept existed. In other words, it may be the case that true classes cannot be efficiently separated in the high-dimensional space by a recursive partitioning method. So, even if it was not possible to identify such global rules accurately predicting recessions, the goal of predicting the latter using some other methods would still exist. For this reason, many statistical and machine learning methods were examined in the framework of this paper. In the previous section we reported the performance of ten different methods based on six datasets and the question is whether some of these methods are consistently better than others.

Before proceeding to answer this last question, it is important to define what the term 'better method' means. Classification Accuracy is an important metric for evaluating different classification methods, but this number alone is not always the only thing we care about. Especially in our problem, which holds that class "0" appears much more often than class "1" due to the rarity of economic recessions, a classifier could possibly achieve more than 80% Classification Accuracy just by correctly predicting only class "0". Such an example is the Logit classifier from Table 7, which achieved Classification Accuracy 84.2%, with Specificity 96.8%, but only 5% in Sensitivity. Apparently, this model should not be used by policymakers for predicting recessions in Mexico, despite its indisputably sound Classification Accuracy. This example makes it clear that, in order to decide which method is better, we must also take into consideration metrics other than Classification Accuracy. In the author's opinion, a model can be considered good in the framework of this paper, if its Classification Accuracy is at least 85%, its Sensitivity and Precision at least 70%, and its False Alarm at most 10%. Of course, this is a reasonable, albeit subjective, choice. In practice, the method to be chosen depends a lot on how much FPR can be tolerated.

Before discussing which methods tend to prevail, it is important to shed some

light on the interpretation of predictions. Taking into account how variable *PreRecess* was constructed, we realise that if a classifier predicts class "1" for a quarter Q_t, it essentially predicts that a recession is going to begin within that year; i.e., in one of the four quarters from Q_t to Q_t+3. This holds because we have focused on pre-recessionary periods rather than on recessions *per se*.

Tables 4 – 9 show the out-of-sample predictive performance of ten methods in six datasets and, at this point, the best ones for each country are presented. We begin our analysis with Table 9 and the USA. For this dataset, the method to be selected seems to be an easy decision. SVM achieved the highest Classification Accuracy (93.45%), having the highest Sensitivity (75%) and one of the lowest values for False Alarm (4.76%). Also, the fact that it has the third highest Precision (73.33%) among all tested models makes SVM a very reasonable choice for predicting recessions in the USA. We move on to Table 4 and the dataset for Australia. In this case, the decision is not as obvious as the previous one. The BRT model achieved the highest Precision (77.78%) with the lowest False Alarm (1.59%), but, despite its good Classification Accuracy (90.7%), its Sensitivity is not adequate (45%). The two models distinguished here are those of SVM and ANN. The latter achieved the highest Sensitivity of the table (88.57%), with Classification Accuracy at 90.12%, but it has the third highest False Alarm (10.08%). On the other hand, SVM has the highest Classification Accuracy of the table (92.44%), with the second highest Sensitivity (86.43%) and its False Alarm is 7.24%. The fact that SVM achieved better Precision than ANN (the third highest of the table: 63% versus 52.06%) is an additional argument favouring the opinion that SVM is the most appropriate method for the case of Australia as well.

Regarding the dataset for Japan (Table 6), the most appropriate model seems to be, once again, that of SVM. It has the highest Classification Accuracy (89.53%), the highest Sensitivity (81.71%) and the second largest Precision (61.35%). Its only weakness seems to be the False Alarm of 12.9%. However, the methods that follow in terms of Classification Accuracy (Random Forests and *k*-NN) – which, additionally, have lower False Alarm – achieved Sensitivity under 40%. So, in order to avoid the cost of missing a lot of true positives, we would rather tolerate some more false positives than those corresponding to our acceptable percentage and, consequently, select the SVM. In the dataset for Germany (Table 5), *k*-NN algorithm achieved the highest Classification Accuracy of the table: 87.5%. However, its moderate Sensitivity (53.13%) discourages us from stating that it can be used for predicting recessions in Germany. The most suitable model here seems to be that of Random Forests. It achieved Classification Accuracy 86.9%, with Sensitivity 71.88% (the highest of the table), Precision 71.88% and False Alarm 8.73%.

Regarding the dataset for Mexico (Table 7), it is not so easy to reach a decision about which method performs best. For a moment, let us ignore the performance of the Ensemble model. In such a table, Decision Trees and Average Trees algorithm

(*K*-fold) show the highest Sensitivity (50%), but they exhibit the two highest False Alarm values (13.02% and 12.23%, respectively). Note the fact that the highest Sensitivity in the dataset for Mexico is only 50%, which disputes its data quality. The BRT model holds the highest Classification Accuracy (85.96%) and the lowest False Alarm (2.76%), but its Sensitivity is only 20%, which makes it seem useless. So, if we ignore the Ensemble model, the best choice for the case of Mexico appears to be the ANN. It has the second largest Classification Accuracy of the table (85.38%), Sensitivity equal to 42.5% (it is the next value after the 50% value mentioned above), the third largest Precision (48.81%) and False Alarm 7.34%. However, since none of these methods alone provided satisfactory results, the idea was to build an ensemble model that makes best use of all three of them. More specifically, the Ensemble model applies the majority rule on the predictions of the Average Trees (*K*-fold), BRT and ANN models[40] in order to make a rather more accurate prediction. Although it achieved the highest Classification Accuracy (85.96%; same as BRT) and the lowest False Alarm of the table (2.55%; even better than BRT), it exhibited only 20% Sensitivity. Therefore, after considering the entire table, it seems that ANN is probably the best choice.

A similar situation exists in the results of the UK dataset (Table 8). The highest value for Sensitivity is 50% and it is achieved only for both variants by the Average Trees algorithm. They provided identical results: Classification Accuracy 88.62%, Precision 41.27% and False Alarm 5.87%. The highest Classification Accuracy (89.82%) is achieved by the *k*-NN algorithm, which also has the lowest False Alarm value (1.4%) and Sensitivity equal to 37.5%. However, the author's decision here was to choose the BRT method. Its model achieved the second largest Classification Accuracy (89.22%), the second highest Sensitivity (43.75%) and the second largest Precision of the table (63.33%). Its False Alarm is 4.47%, which seems quite acceptable. However, we should not neglect the fact that it was not possible for the UK dataset, either, to find a model that achieved above 50% Sensitivity with good Classification Accuracy. For this reason, an ensemble model was constructed for the UK, too, but its results were identical to those of Average Trees. Thus, this country is another case for which it was not possible to find a satisfactory model. If this did not happen because of poor data quality or because of omitting some important variables, then it may be the case that, for some methods, the correct set of parameters cannot not be found. As it is computationally infeasible to search through all possible combinations among model parameters (e.g. C and γ in the SVM framework), it is likely that we missed the opportunity to fit a better model for Mexico and the UK due to computational restrictions. Table 10 shows which methods were selected by the author for each country:

40. Their parameterisation remained unchanged.

Table 10. The methods selected per country.

Country	Method	Classification Accuracy	Sensitivity	Precision	Specificity	False Alarm	F_1-Score
AUS	SVM	92.44%	86.43%	63%	92.76%	7.24%	70.74%
GER	Random Forests	86.9%	71.88%	71.88%	91.27%	8.73%	68.54%
JAP	SVM	89.53%	81.71%	61.35%	87.1%	12.9%	75%
MEX	ANN	85.38%	42.5%	48.81%	92.66%	7.34%	46.21%
UK	BRT	89.22%	43.75%	63.33%	95.53%	4.47%	73.33%
USA	SVM	93.45%	75%	73.33%	95.24%	4.76%	71.43%

The mean performance of each method is presented in summary in the two tables of Figure 7:

Out-of-sample			
Method	Accuracy	Sensitivity	False Alarm
Average Trees (K-fold)	83.82%	45.28%	8.99%
Average Trees (random)	85.19%	45.25%	7.67%
Decision Trees	83.72%	36.01%	6.96%
Random Forests	87.54%	43.34%	5.58%
Logit	86.65%	31.53%	4.56%
Probit	86.56%	29.86%	4.63%
k-NN	88.13%	44.53%	4.10%
BRT	87.44%	38.44%	4.08%
SVM	88.50%	56.25%	7.77%
ANN	83.62%	54.45%	13.25%

In-sample			
Method	Accuracy	Sensitivity	False Alarm
Average Trees (K-fold)	92.81%	68.73%	3.74%
Average Trees (random)	91.05%	72.20%	6.30%
Decision Trees	92.23%	65.61%	2.84%
Random Forests	99.60%	97.74%	0%
Logit	88.50%	36.19%	3.19%
Probit	88.30%	33.96%	3.07%
k-NN	95.07%	76.08%	2.09%
BRT	93.50%	70.98%	3.66%
SVM	81.39%	58.12%	16.24%
ANN	85.11%	59.92%	12.86%

Figure 7. Mean performance of each method in terms of Classification Accuracy, Sensitivity and False Alarm.

The same information is presented in the bar chart of Figure 8. From these two figures we realise that SVM had the best out-of-sample performance on average, in terms of Classification Accuracy and Sensitivity. Moreover, we see that the Logit and Probit models showed very poor performance in terms of Sensitivity. The Average Trees algorithm (especially the variant that performs random sampling) provided better results than classic Decision Trees and, on average, it outperformed ANN in terms of Classification Accuracy. As for the in-sample performance, we observe that the Random Forests models tend to perfectly overfit their training data.

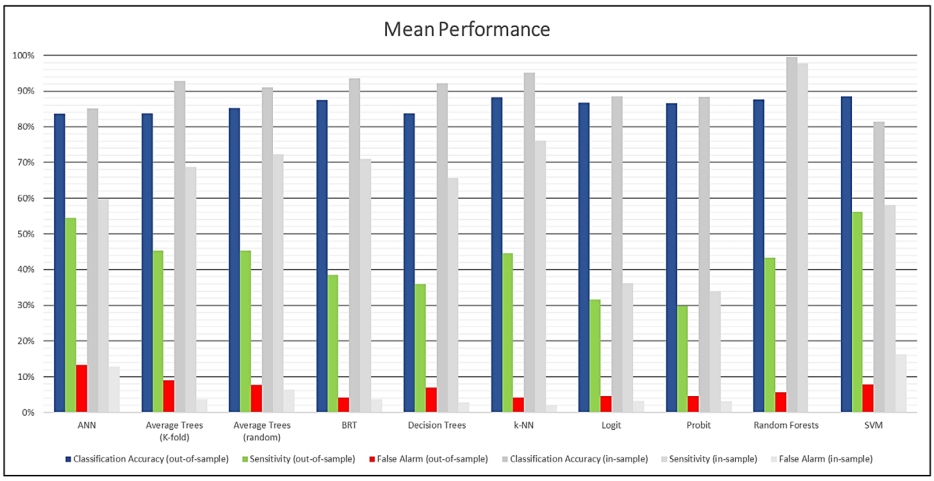

Figure 8. Mean performance of each method.

Australia | Method: SVM						
Horizon	Classification Accuracy	Sensitivity	Precision	Specificity	False Alarm	F1-Score
3 months	87.50%	100.00%	21.05%	87.07%	12.93%	34.78%
6 months	88.24%	100.00%	22.22%	87.83%	12.17%	36.36%
1 year	88.89%	100.00%	23.53%	88.50%	11.50%	38.10%
2 years	91.15%	100.00%	28.57%	90.83%	9.17%	44.44%

Mexico | Method: ANN						
Horizon	Classification Accuracy	Sensitivity	Precision	Specificity	False Alarm	F1-Score
3 months	83.19%	12.50%	25.00%	94.17%	5.83%	16.67%
6 months	83.05%	18.75%	30.00%	93.14%	6.86%	23.08%
1 year	81.90%	12.50%	22.22%	93.00%	7.00%	16.00%
2 years	83.04%	15.38%	20.00%	91.92%	8.08%	17.39%

Germany | Method: Random Forests						
Horizon	Classification Accuracy	Sensitivity	Precision	Specificity	False Alarm	F1-Score
3 months	91.45%	75.00%	81.82%	95.70%	4.30%	78.26%
6 months	84.48%	45.83%	68.75%	94.57%	5.43%	55.00%
1 year	80.70%	25.00%	60.00%	95.56%	4.44%	35.29%
2 years	75.45%	8.33%	28.57%	94.19%	5.81%	12.90%

UK | Method: BRT						
Horizon	Classification Accuracy	Sensitivity	Precision	Specificity	False Alarm	F1-Score
3 months	77.59%	33.33%	18.18%	82.69%	17.31%	23.53%
6 months	80.00%	33.33%	21.05%	85.44%	14.56%	25.81%
1 year	76.99%	41.67%	20.83%	81.19%	18.81%	27.78%
2 years	67.89%	41.67%	15.15%	71.13%	28.87%	22.22%

Japan | Method: SVM						
Horizon	Classification Accuracy	Sensitivity	Precision	Specificity	False Alarm	F1-Score
3 months	80.88%	80.00%	64.00%	81.25%	18.75%	71.11%
6 months	73.13%	60.00%	54.55%	78.72%	21.28%	57.14%
1 year	64.62%	50.00%	43.48%	71.11%	28.89%	46.51%
2 years	68.85%	63.16%	50.00%	71.43%	28.57%	55.81%

USA | Method: SVM						
Horizon	Classification Accuracy	Sensitivity	Precision	Specificity	False Alarm	F1-Score
3 months	88.89%	75.00%	35.29%	89.91%	10.09%	48.00%
6 months	88.79%	62.50%	33.33%	90.74%	9.26%	43.48%
1 year	82.46%	62.50%	22.73%	83.96%	16.04%	33.33%
2 years	87.27%	62.50%	31.25%	89.22%	10.78%	41.67%

Figure 9. The forecasting performance of the methods selected per country, at four horizons.

In Figure 9, we can assess the performance of the methods selected for different forecasting horizons. Parameterisation remained the same for each model. SVM performed very well for Australia and the USA, while it showed moderate increase in False Alarm for larger forecasting horizons in the case of Japan. Moreover, we observe that the Random Forests model for Germany loses its ability to correctly predict an upcoming recession for horizons longer than six months. Instead, the SVM models seem more robust in increasing the forecasting horizon order. For the case of Australia, we observe perfect performance of SVM in terms of Sensitivity. Since the parameterisation of the SVM model (γ and C) was defined from the same data during

the training phase, it is very likely that an effect of overfitting exists here, despite the fact that forecasting at horizons comprises, by definition, a type of out-of-sample performance evaluation. In general, we could avoid this situation by leaving a specific set of observations out of any training procedure. However, in our framework this would not be a wise decision because recessions are rare events and such a test set probably would barely contain even one pre-recessionary period. In our analysis, every pre-recessionary period appeared at least once in a test set for every model. Lastly, for the cases of Mexico and the UK, the situation remained, more or less, the same.

The second question was whether any method prevails in predicting recessions. From the analysis above it seems that SVM tends to do so. It was chosen for three out of six countries, while no other method was chosen more than once. It probably has the potential to perform better for the other three countries, too, if a more suitable combination of C and γ is found. Nevertheless, the trial and error procedure on the SVM parameter setting could not provide better out-of-sample performance for those countries. As for the Average Trees algorithm, it is true that its results are simple and easily interpretable. However, given that a recession is going to start within next year, it seems that it is not possible to predict the upcoming recession in half of such cases by using this method[41]. Despite the fact that such simple rules about macroeconomic variables cannot predict economic recessions in an accurate way[42], after this analysis we can argue that methods providing fewer insights from an economic perspective, such as SVM or Random Forests, can be proven useful for policymakers because of giving correct early warning signs in the majority of the cases.

Another question was whether a set of general rules that lead to recessions exists, at least for the countries studied. From the findings of this paper the answer is *no*. Important variables differ among countries, so pre-recessionary conditions are necessarily different according to our findings. We can argue that there are some *country-specific* macroeconomic conditions often preceding recessions – this is the output the Average Trees algorithm provides – but we cannot support the opinion that national economies operate in a way that consistently follows such simple rules. At a second glance, we could say that these rules – if they exist – need to be represented by a more complex concept than a tree-like one.

Closing this section, we must also answer the question whether there is any economic theory verified through the findings of this paper. In order to give an answer to this question, we must look at the set of variables selected for each one of the six countries. Each economic theory selected from relevant literature is represented in our analysis through certain variables. Assuming our data contain no systematic errors,

41. This conclusion arises from the method's performance in Sensitivity.
42. An example of Average Trees algorithm visual output is presented in Appendix, A.6.

and they are what they are supposed to be[43], we expect that, if a theory is consistent with reality, its variables tend to be characterised as important by the procedure applied. So, by looking at the relevant plots (Figures 1–6), we realise that in all countries except Germany there is always a debt-related variable in the set of the ones selected. For the case of Germany, the first such variable is two positions under the red line – not so far from being chosen, though. The remaining variables appear less frequently in the sets of the ones selected. Debt-related variables were introduced in the datasets due to I. Fisher's debt-deflation theory. Can we say that this is the theory which best corresponds to reality? The answer is *maybe*. While we could agree with Fisher that factors related to debt have an important impact on the evolution of business cycles[44], we completely miss the 'deflation' part in our analysis. In the context of this paper, deflation means $GrowCPI < 0$. Apart from the USA, there is no other country for which an inflation-related variable is characterised as important. Therefore, we cannot say that the debt-deflation theory is truly verified through this paper. Alternatively, we could say that it will be a rather beneficial decision to include debt-related variables in similar future works.

Concluding remarks and topics for further research

This paper was an opportunity for investigating the topic of economic recession forecasting from a new basis. Instead of incorporating in our analysis only the variables most papers in literature suggest, we started our investigation from point zero. The rationale for choosing the main variables presented stems from theories established very early on in economic science. Through this work it was made possible to test these theories against each other and discover which of their hypotheses are confirmed by data. Moreover, we reviewed some recent papers from relevant literature in order to present the kind of methodologies applied today and to find which additional variables may potentially make good predictors.

One innovation of this paper is focusing on the short period before a recession begins and not on the recession *per se*. The advantage of this choice is that the predictions resulting from it refer to potentially pre-recessionary periods. This means it is very likely that if a policymaker takes them into account, they have the time to design a proper policy and, ultimately, intervene in the economy. If predictions referred to recessionary periods, it would probably be too late for a policymaker to take precautionary measures. On the other hand, though, many explanatory variables behave in a known much clearer manner during recessionary periods than in the last quarters

43. For example, *BCI* is a variable that indeed sufficiently captures the expectations businesspeople have.
44. We found that models based on such variables can predict pre-recessionary periods with adequate accuracy, at least for three out of the six countries (Australia, Japan and the USA).

prior to a recession. For example, unemployment may rise even from the onset of a recession. Thus, following the conventional approach translates to less uncertain predictions. However, in this paper, the choice was to detect early recessionary signs, even allowing for some false alarms. An additional issue that refers to the whole methodology is that of data availability. For us to be able to make a timely prediction, it is important to have updated necessary data, as far as this is possible. However, there are many variables – at least in the OECD database – that are published once a year, which makes such a goal more challenging.

Probably the most important innovation of this paper is the Average Trees algorithm. It generally achieved better out-of-sample performance than classic Decision Trees, while its good interpretability remained unchanged. It is characterised as most important because it provides us with an alternative way to extract decision tree rules, which are very likely more generalisable than classic ones. However, we should not neglect the contribution of all methodological steps before building average trees. Initially, for every dataset more than 100 variables existed, while there were roughly 170 observations. The number of variables had to be significantly smaller, in order to build parsimonious, yet well-generalisable, models in the steps to follow. The variable selection procedure through fitting an initial Random Forests model with 10,000 trees was proved efficient; models of different methods based on only 'important variables' showed very good out-of-sample performance in evaluation metrics. Even the inclusion of lagged variables and variables of percentage changes and differences has proved helpful, as numerous such variables were above the red line in variable importance plots. Furthermore, the construction of some new variables seemed to be a correct decision in the attempt to build well-generalisable models. And even if it was somehow expected for variable *Spread*, as we know that many researchers take it into account in similar works, it was probably not expected for variable *ElastRC* from the Marxian analysis[45]. For the case of the USA, these two variables were quite enough to predict more than half of the recessions using the Average Trees algorithm (see Appendix, A.7). Therefore, although Average Trees did not show better out-of-sample performance than Random Forests, it seems that the overall dataset preparation procedure had a positive impact on evaluation results in general.

We saw that for four out of six countries it was possible to find models with satisfactory performance. This is a good sign for the methodology developed, but six datasets cannot probably be considered a sufficiently large number. A suggestion for further research is to repeat the same methodological steps for many more countries and evaluate them in the same manner. Thus, it could probably be possible to draw more certain conclusions regarding the effectiveness of the methodology applied.

45. This is said because it was not possible for the author to find a quantitative study that included a variable identical – or similar – to *ElastRC* in models aiming at the prediction of economic recessions.

Additionally, it would probably be possible to give more certain answers about which theory seems more plausible. Moreover, testing some additional methods may lead us to find satisfactory models for the two countries this was not achieved (i.e., Mexico and the UK). Lastly, probably the most interesting aspect of investigating this topic in a future research project is through a complex network framework. In this paper we completely missed the dimension of interconnectivity among countries. Each country was studied as if it were isolated from the rest of the world. However, economies are interconnected and a recession in country *A* may cause a recession in country *B* after some time. In order to build a realistic model for predicting economic recessions, one should look at this aspect, too, because the global economy does operate as a single system. In fact, countries are not isolated from each other. In this paper it was not feasible to follow such an approach, since it essentially requires training and testing complex network models that encompass dozens of countries. In a more extensive future research, though, it is very likely that such an approach will provide us with even better models, because it may incorporate a very important aspect of business cycles evolution; namely, that of interconnectivity among national economies.

Appendix

A.1 Details about the methods mentioned

With regard to the Logit and Probit models, details can be found in Baltagi (2002, pp. 332-333). Regarding Support Vector Machines (SVM), a very good presentation of the method can be found in James *et al.* (2013, pp. 341-353). The *k*-NN algorithm can be found in Kubat (2017, p. 44). Decision Trees and Random Forests are presented in James *et al.* (2013, pp. 304-313), and (2013, pp. 319-321), respectively. Artificial Neural Networks (ANN) are extensively presented in Bishop (2006, pp. 225-231). Finally, the Boosted Regression Trees (BRT) algorithm can be found in James *et al.* (2013, p. 323).

A.2 Data references

1) *BankRate*
   Bank of England (2018), Official Bank Rate History Data from 1694. https://www.bankofengland.co.uk/monetary-policy/the-interest-rate-bank-rate (Accessed on 10 June 2018)

   Bank of Japan (2018), The Basic Discount Rates and Basic Loan Rates. https://www.boj.or.jp/en/statistics/boj/other/discount/index.htm/ (Accessed on 10 June 2018)

   Bank of Mexico (2018), Representative interest rates. http://www.banxico.org.mx/SieInternet/consultarDirectorioInternetAction.do?accion=consultarCuadroAnalitico&idCuadro=CA51&sectorDescripcion=Precios&locale=en (Accessed on 14 June 2018)

   Deutsche Bundesbank (2018), Discount rate, Lombard rate and base rate. https://www.bundesbank.de/Navigation/EN/Statistics/Money_and_capital_markets/Interest_rates_and_yields/Tables/table.html (Accessed on 10 June 2018)

   FRED Economic Data (2017), Interest Rates, Discount Rate for United States. https://fred.stlouisfed.org/series/INTDSRUSM193N (Accessed on 10 June 2018)

   Reserve Bank of Australia (2018), Cash rate. https://www.rba.gov.au/statistics/cash-rate/ (Accessed on 10 June 2018)

2) *BCI*
   OECD (2018), Business confidence index (BCI) (indicator). doi: 10.1787/3092dc4f-en (Accessed on 05 June 2018)

3) *CPI*
   OECD (2018), Inflation (CPI) (indicator). doi: 10.1787/eee82e6e-en (Accessed on 04 June 2018) (Accessed on 05 June 2018)

4) *Fdebt*
OECD (2018), Financial corporations debt to equity ratio (indicator). doi: 10.1787/a3108a99-en (Accessed on 08 June 2018)

5) *Fprof*
OECD (2018), Value-added in financial corporations (indicator). doi: 10.1787/f891bfeb-en (Accessed on 09 June 2018)

6) *Gdebt*
OECD (2018), General government debt (indicator). doi: 10.1787/a0528cc2-en (Accessed on 08 June 2018)

7) *GFCF*
OECD (2018), Investment (GFCF) (indicator). doi: 10.1787/b6793677-en (Accessed on 05 June 2018)

8) *Gov*
OECD (2018), General government spending (indicator). doi: 10.1787/a31cbf4d-en (Accessed on 08 June 2018)

9) *GrowGDP*
OECD (2018), Quarterly GDP (indicator). doi: 10.1787/b86d1fc8-en (Accessed on 05 June 2018)

10) *HHC*
OECD (2018), Household spending (indicator). doi: 10.1787/b5f46047-en (Accessed on 06 June 2018)

11) *HHdebt*
OECD (2018), Household debt (indicator). doi: 10.1787/f03b6469-en (Accessed on 08 June 2018)

12) *LIR*
OECD (2018), Long-term interest rates (indicator). doi: 10.1787/662d712c-en (Accessed on 08 June 2018)

13) *M*1
OECD (2018), Narrow money (M1) (indicator). doi: 10.1787/7a23d68b-en (Accessed on 06 June 2018)

14) *Mports*
OECD (2018), Trade in goods and services (indicator). doi: 10.1787/0fe445d9-en (Accessed on 06 June 2018)

15) *NFdebt*
OECD (2018), Non-Financial corporations debt to surplus ratio (indicator). doi: 10.1787/dc95ffa7-en (Accessed on 08 June 2018)

16) *NFprof*
OECD (2018), Value-added in non-financial corporations (indicator). doi: 10.1787/731f0874-en (Accessed on 09 June 2018)

17) *Pop*
OECD (2018), Population (indicator). doi: 10.1787/d434f82b-en (Accessed on 08 June 2018)

18) *PPI*
OECD (2018), Producer price indices (PPI) (indicator). doi: 10.1787/a24f6fa9-en (Accessed on 05 June 2018)

19) *PPP*
OECD (2018), Purchasing power parities (PPP) (indicator). doi: 10.1787/1290ee5a-en (Accessed on 08 June 2018)

20) *RnD*
OECD (2018), Gross domestic spending on R&D (indicator). doi: 10.1787/d8b068b4-en (Accessed on 08 June 2018)

21) *Sav*
OECD (2018), Saving rate (indicator). doi: 10.1787/ff2e64d4-en (Accessed on 06 June 2018)

22) *SIR*
OECD (2018), Short-term interest rates (indicator). doi: 10.1787/2cc37d77-en (Accessed on 08 June 2018)

23) *SPI*
OECD (2018), Share prices (indicator). doi: 10.1787/6ad82f42-en (Accessed on 05 June 2018)

24) *Tax*
OECD (2018), Tax revenue (indicator). doi: 10.1787/d98b8cf5-en (Accessed on 08 June 2018)

25) *Unemp*
OECD (2018), Unemployment rate (indicator). doi: 10.1787/997c8750-en (Accessed on 06 June 2018)

26) *Wage*
OECD (2018), Average wages (indicator). doi: 10.1787/cc3e1387-en (Accessed on 07 June 2018)

27) *Xports*
OECD (2018), Trade in goods and services (indicator). doi: 10.1787/0fe445d9-en (Accessed on 06 June 2018)

A.3 Pre-processing steps

To facilitate reproducibility of this paper's results, in this section we present all steps followed to build the final dataset for each country from the raw data downloaded. The data in OECD's database are grouped per variable. This means that each file downloaded referred to one variable[46] and contained data about *all* countries available. Therefore, the first step was to create six .CSV files – one per country – for each main variable (except for *BankRate*). This simple step was executed in a spreadsheet. Regarding the *BankRate* variable, which is the only one not available in OECD's database, a different procedure was followed. Data about *BankRate* are officially provided by central banks. Each country's central bank makes an announcement when a change in its bank rate is to take place, but these announcements have no specific frequency. Moreover, there are differences in file format[47] and/or data structure among the data published by each central bank. This means that, while for the rest of the variables it was possible to import values into an initial data frame in an automated way (as indeed happened), for the *BankRate* this was not possible; or, at least, it was not worth the effort to build such a complicated procedure just for one variable. Consequently, the data concerning *BankRate* were entered into the initial data frames manually[48].

After creating the .CSV files, everything was ready for the construction of the initial data frames, i.e., this very first version of the six datasets before the main pre-processing steps. At this point, there was a folder for each country, which contained as many .CSV files as the number of that country's main variables. Initial data frames were built using the script[49] Building Datasets.R, in RStudio. This script was written in order to transfer all data from the .CSV files into the six data frames quickly and accurately. Data from variables of quarterly frequency were just copied one-by-one, because they already had the frequency desired. Data from yearly variables were entered into the first quarter (Q1) of each year, without filling the empty cells yet. Variable *BCI* was the only one of monthly frequency. For this variable, a three-month average had already been computed in the spreadsheet for every available quarter. These averages were entered into the corresponding quarters of the initial data frames using the Building Datasets.R script, as well. Finally, *BankRate* was the only variable that was entered into the initial data frames manually, as already mentioned.

---

46. An exception was the "Trade in goods and services" indicator, which contained data about both imports and exports – two different variables in our datasets.
47. For example, Deutsche Bundesbank publishes these data in .PDF format, while the other five central banks provide them in either .XLS or .CSV format.
48. As these data were not provided in quarterly format, a weighted average of the corresponding bank rates was calculated for each quarter. For example, if a bank rate changed from a to b in the middle of a quarter, a simple average of these two bank rates was entered into the initial data frame for that quarter. In fact, the weights varied a lot. As this procedure was accomplished in a spreadsheet by hand, it is expected that there are some small deviations from the actual weighted quarterly averages.
49. All the script files mentioned, along with the data downloaded, are available to anyone interested by request via email.

At that point we had six datasets with raw data and many missing values. The goal of the next steps was to fully prepare the datasets for model fitting. These are the main pre-processing steps. For this purpose, several functions were written in R by the author, in order to make these pre-processing steps a fully automated procedure. These functions can be found in the Preprocessing.R script, which is the same file for all countries. The main function of this script is the preprocess(dt), where dt is an argument that represents a data frame. The input of this function is a dataset which contains only raw data, as described above, and the output is the final version of that dataset. The preprocess() function is based on other functions that perform the main pre-processing steps, i.e., its only purpose is to coordinate the entire process. These functions are the following:

a) q1.q4(dt): This function selects the variables of yearly frequency and transfers their observations from Q1 to Q4. This is a preparatory step for the transformation of yearly variables into quarterly ones. The rationale of this step lies behind the fact that an observation of yearly frequency contains information about all four quarters. In other words, information about Q4 cannot, in fact, be known in Q1–Q3 and, therefore, the best probable choice is to transfer the yearly observation into Q4. This choice is still a convention that stems from our lack of further information.
b) def.recess(dt): This function captures the pre-recessionary periods. In order to find them, it checks only the values of *GrowGDP*. The definition used is that a recession is *a period of two consecutive quarters of decline in real GDP*. Each time the function finds such a period, it labels the first recessionary quarter and the previous three with number "1". In contrast to the common practice of labelling the recessionary periods with number "1", in this paper we label the pre-recessionary periods as such, because what concerns us is the conditions *before* a recession. The only reason we mark the first recessionary quarter as well is because a recession might, in fact, have begun some time during the quarter, and, therefore, part of that quarter is also a pre-recessionary period. The remaining periods are marked with number "0" and a new binary variable is imported into the dataset, namely, the *PreRecess*. It now becomes clear how a classification method could capture the conditions before the beginning of a recession.
c) interpolate(dt): This is the first function that deals with the issue of missing values. It ignores the missing values before the first real observation (there is another function concerning them) and it completes the remaining missing values by applying linear interpolation, where this is feasible. Essentially, interpolate() is applied to variables of yearly frequency[50]. If, for example, a variable has the

50. The function does *not* make a distinction between yearly and quarterly variables, but, in practice, it intervened only in variables of yearly frequency due to the way these were entered into the dataset.

values 1990.Q4: 100, 1991.Q1: *NA*, 1991.Q2: *NA*, 1991.Q3: *NA* and 1991.Q4: 500, the function fills the three missing values with the numbers 200, 300 and 400, respectively. The main problem of all yearly variables is the fact that their missing values appear in a systematic way: after every observation, three missing values necessarily follow. For these variables, linear interpolation is a simple and a rather reliable solution to this problem. The only implicit assumption is that the dots (i.e., the actual observations) are connected by a straight line; in the face of uncertainty, there is no reason to assume something more complicated.

d) new.vars(dt): This function creates two new variables in the dataset. One is *Spread*, which refers to the interest rate spread, and the other is *ElastRC*, which refers to the elasticity of profit rate w.r.t. capital ($e_{r,c}$). The transformations were the following: $Spread_t = LIR_t - SIR_t$ and $ElastRC_t = \frac{(Fprof_t+NFprof_t)-(Fprof_{t-1}+NFprof_{t-1})}{GFCF_t*(Fprof_{t-1}+NFprof_{t-1})}$, where *t* indicates time (quarter of year). With *ElastRC* we want to approximate quantity $\frac{dr}{dC}\frac{C}{r}$ from the analysis in Tsoulfidis (2010, pp. 119-120). By definition, $GFCF = \frac{dC}{C}$, it is the growth rate of capital formation. For the calculation of $\frac{dr}{r}$, we approximate[51] total profit rate with quantity $r \cong Fprof + NFprof$. Therefore, we approximate quantity $\frac{dr}{r}$, by calculating: $\frac{r_t-r_{t-1}}{r_{t-1}} = \frac{\Delta r}{r}$. Putting everything together, we end up with the transformation given above for *ElastRC*.

e) imput.NAs(dt): This is the function that handles the remaining missing values. With function interpolate() we have filled all missing values between the first and the last actual observation of every variable. However, interpolation is not feasible for the missing values before the first actual observation and after the last one. This is the task of imput.NAs() function. The following steps are a summary of its main steps on each variable with many missing values[52]:

51. Variables *Fprof* and *NFprof* are indicators of profitability. However, according to the full definitions provided by OECD (references in Appendix, A.2), they are not constructed as profit-to-capital ratios and this is why the verb 'approximate' is used here.
52. We allow up to sixteen missing values per variable, in the total of the 196 possible observations. If there are more than sixteen, then the function proceeds to missing data imputation. Even without this threshold, imput.NAs() cannot complete all missing values in general, because the imputation procedure is based on observations of other variables. In other words, the output of the function is always restricted by the number of other variables' missing values. Therefore, in order to save computational time (i.e., to avoid running the whole procedure just to replace "*NA*" with "*NA*"), we allow a relatively small number of missing values to exist in datasets.

1) The first step is to find the twenty variables that are the ones most correlated with the selected one. A correlation matrix for all variables of the dataset has already been computed. At this point it is important to mention that, where it was feasible, lagged variables and variables of first-order differences or percentage changes have been calculated for every main variable before running the imput.NAs() function. This means that, at the time imput.NAs() runs, the total number of variables is by far larger than 100 for all countries. This is the reason why only twenty variables were selected. The functions that created those extra variables are grows.diffs(dt) and L1.L2(dt), which are presented at a later point in this discussion.
2) Variables with more than sixteen missing values are excluded from the set of twenty. The reason becomes apparent in the next step.
3) A linear model is fitted, using as a response variable the one that needs imputation, and as predictors the remaining variables from the set of twenty. Since these remaining variables are going to be used as predictors in a linear model, they should not have many missing values[53]; this is the explanation for the previous step. Function step() is used to build a parsimonious linear model, using BIC as model selection criterion. Argument direction is set to "forward", so a variable is added only if it reduces the BIC value.
4) Using the fitted model, imput.NAs() estimates the missing values of the variable selected, where this is possible. In order to reduce a potential high variance of these estimates, a smoother is applied on them before completing the missing values in the dataset. Again, in the face of uncertainty, the preference here is to capture the (more certain) trend-like movements rather than the (uncertain) data noise. Additionally, sometimes it may be the case that this procedure produces negative values for a variable that takes only non-negative values; e.g. variable *Wage*. Function imput.NAs() addresses this issue by properly "squeezing" all estimates into the positive range. To be more specific, when such a violation is verified, the minimum estimate (which is negative) takes the value of 0.001 (this is a convention) and all others are proportionally moved towards the first actual observation of the variable[54]. Thus, estimates close to the first *actual* observation are subject to minor changes, while further estimates do change more. This seems a good approach for coping with the problem of wrongly negative estimates, because, on the one hand, the correlation between the response variable and its predictors is maintained (as the estimates' transformation is linear)

53. The number of missing values that imput.NAs() will not finally complete, equals that of the predictor with the most missing values.
54. In mathematical terms, the transformation is the following: $v_i' = v_i + \alpha\frac{\beta - v_i}{\gamma} + 0.001$, where v_i is the i-th estimate, $\alpha = \left|\min_{i:\{1,2\ldots\}} v_i\right|$, β is the first actual observation of the selected variable, and $\gamma = \alpha + \beta$.

and, on the other hand, the non-negative nature of this variable[55] is respected. An evaluation of the imput.NAs() method through an example can be found in Appendix, A.4.

f) grows.diffs(dt): This function constructs the variables of percentage changes and the variables of first-order differences. Including such variables in our analysis allows us to examine the main problem of this paper from a dynamical perspective as well. Each variable, which is *not* a percentage or a ratio, has a counterpart variable with the prefix *Grow* before its main name. This indicates a variable that refers to percentage changes. For example, *BCI* is an index, i.e., not a percentage or ratio. Thus, a variable *GrowBCI* is also included in the six datasets, and is equal to $\frac{BCI_t - BCI_{t-1}}{BCI_{t-1}}$x100. The reason for using percentage changes in these cases, and not merely differences, is because we want to provide models with variables that have comparable values over time. For example, an increase in average income by €1,000 today does not create the same buying power, as it could have done 50 years ago, due to inflation. The use of percentages for describing such changes makes comparison much more meaningful. For variables that are already percentages or ratios, their first-order differences are calculated in their counterpart variables with the prefix *Dif*. For example, variable *Tax* is expressed as a percentage of GDP. Therefore, in this case the variable $DifTax = Tax_t - Tax_{t-1}$ was constructed.

g) L1.L2(dt): This function constructs variables of first and second lag order for every variable of the dataset (i.e., also for *Grow*-s and *Dif*-s). Exceptions are the variables *PreRecess*, *BCI*, *CPI*, *M*1, *Pop*, *PPI*, *SPI* and *Wage*. The first one is a binary variable. The reason for excluding the rest is that these variables are of no interest at their levels, because their values are not comparable over time. Instead, their *Grow*-s and *Dif*-s variables are entered on to the function L1.L2() for constructing the corresponding lagged variables.

h) remain.Lags(dt): This function recalculates all lagged variables. After imputing the missing data, it is necessary to update the lagged variables with the new values.

---

55. An alternative approach applied in order to cope with the problem of wrongly negative estimates was to use glm() instead of lm() for the linear model, with argument family set to Gamma. This approach succeeded in giving only positive estimates. However, regardless of the variable selected, almost all estimates produced were slightly above zero. Thus, this approach was not adopted, since it failed to produce realistic estimates.

To put everything together, the function preprocess() constructs a dataset as follows:

Algorithm A.1 The main pre-processing steps

INPUT: dt // A data frame with raw data.

1: $dt \leftarrow q1.q4(dt)$
2: $dt \leftarrow def.recess(dt)$
3: $dt \leftarrow interpolate(dt)$
4: $dt \leftarrow grows.diffs(dt)$
5: $dt \leftarrow L1.L2(dt)$
6: $dt \leftarrow imput.NAs(dt)$
7: $dt \leftarrow new.vars(dt)$
8: $dt \leftarrow grows.diffs(dt)$
9: $dt \leftarrow remain.Lags(dt)$

OUTPUT: dt

A.4 Evaluating the imput.NAs() method

There are many R packages the objective of which is reliable missing data imputation. Before creating the function imput.NAs(), such a package was tested on the datasets of this paper in order to decide whether it was worth the effort to create a new procedure for this purpose or to let an existent method complete the remaining missing values. This package was the missForest. At first glance, it seemed that function missForest() could not capture time trends in any variable. Thus, the author's decision was to create function imput.NAs() in order to produce more realistic estimates of missing values. In order to justify this choice, a comparison was made between the two functions in terms of Mean Squared Error (MSE) on known data[56]. The two functions were tested on the initial dataset of the USA. Eight variables with fewer than sixteen missing values were randomly selected, and a range of observations was defined, within which no value was missing from the dataset[57]. Then, 50 missing values were artificially created on these eight variables, in order to simulate a real situation, when some variables present a lot of missing values. The goal was to see how well each function 'predicted' the 50 missing values.

56. The relevant code is provided in the script Miscellaneous.R.
57. This range was 15th observation – 184th observation.

	missForest() MSE	imput.NAs() MSE	Difference
HHC	2.269	0.192	2.077
Mports	1.830	0.385	1.445
Tax	0.662	0.649	0.013
M1	200.651	24.221	176.430
PPI	188.876	10.189	178.668
NFprof	0.084	1.959	-1.875
SPI	125.146	5.928	119.218
Unemp	1.231	1.062	0.169

Figure A.1 Comparison between missForest() and imput.NAs() in terms of MSE.

In Figure A.1 we see that, in seven out of eight variables, function imput.NAs() performed better. In Figure A.2, we can visually assess the performance of the two functions on the eight variables selected. Black lines represent real data, red lines represent estimates of missForest() and green lines represent estimates of imput.NAs().

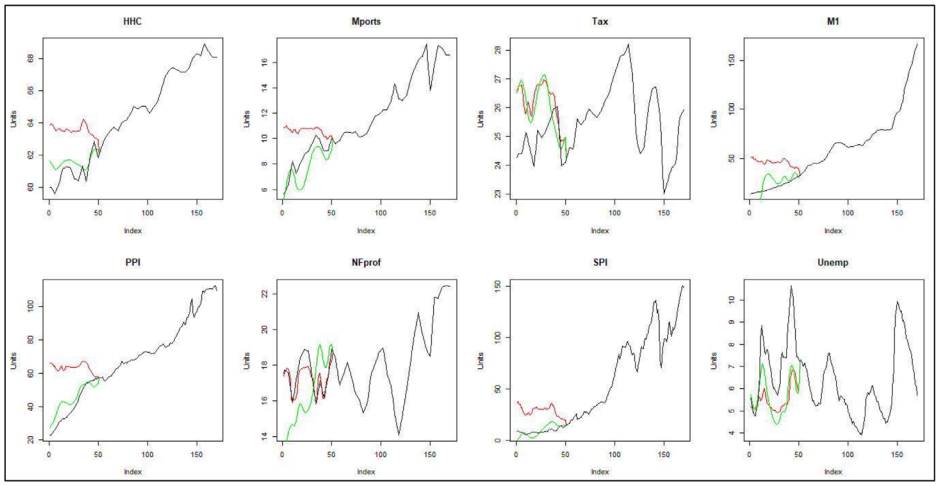

Figure A.2 Optical evaluation of missForest() and imput.NAs() performance (red and green lines, respectively).

Consequently, function imput.NAs() was selected for completing the remaining missing values of the six datasets, since it seems to produce more reliable estimates. This does not imply that imput.NAs() is a generally better imputation method than missForest(); we simply have evidence that it performs better in this specific kind of datasets and, therefore, it was finally used. At this point we must note that the performance of imput.NAs() was dramatically improved when the lagged variables and

the variables of first-order differences or percentage changes were calculated before the missing values imputation. Initially, the plan was to calculate these variables as a last step; however, having calculated all of them before running the imput.NAs() function improved the latter's performance to a large extent.

A.5 Searching for optimal parameter setting in SVM (example)

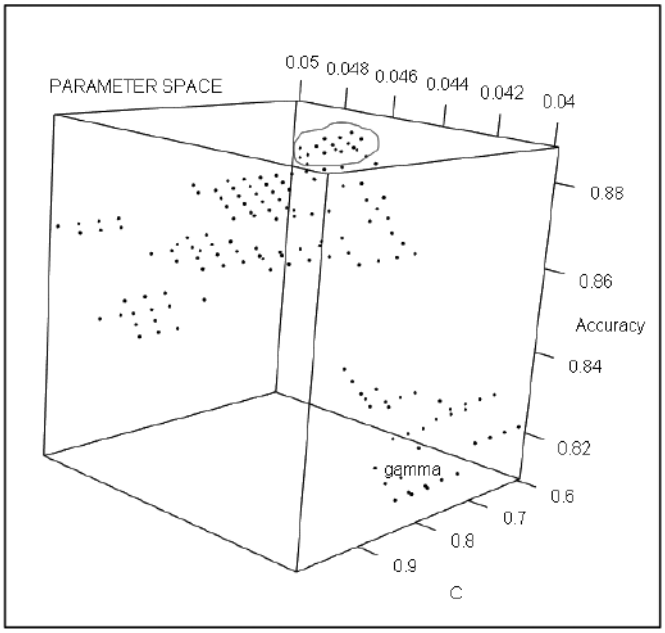

Figure A.3 An example of the three-dimensional plot used for finding the combination of C and γ that gives the best out-of-sample Classification Accuracy.

A.6 The visual output of the Average Trees algorithm (example)

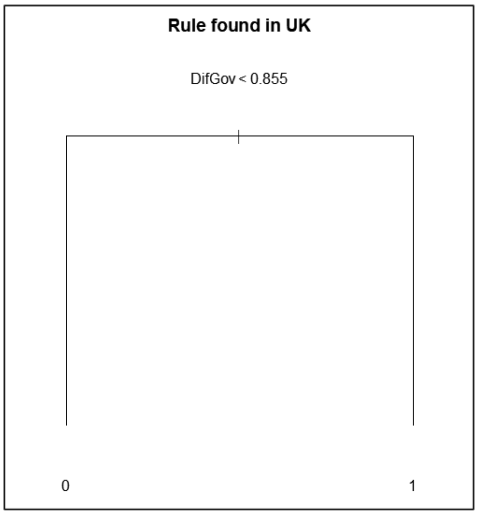

Figure A.4 The result of Average Trees (random sampling) using all 167 observations of the UK dataset (the model parameterisation is presented in Table 8). In this example, a recession is expected within next year provided the current quarter holds that $DifGov \geq 0.855$.

A.7 The result of Average Trees algorithm for the case of the USA

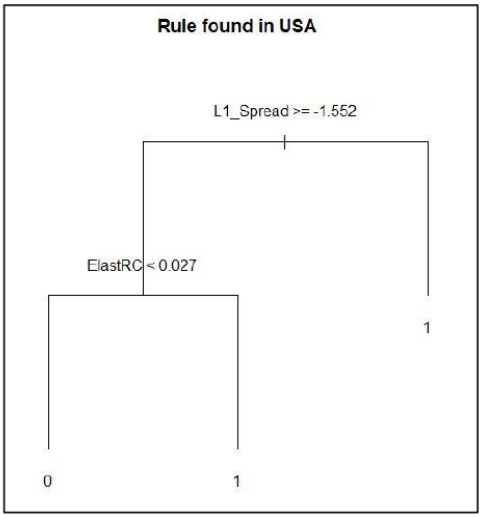

Figure A.5 The output of Average Trees (random sampling) for the dataset of the USA. Regarding the splitting point (inequality), answer "Yes" is on the left-hand side, as usual. The parameterisation of this model is presented in Table 9.

A.8 Evaluation results: In-sample performance

Table A.1 Australia

Method	Classification Accuracy	Sensitivity	Precision	Specificity	False Alarm	F_1-Score	Comments
Average Trees (K-fold)	96.51%	77.14%	91.67%	99.33%	0.67%	93.08%	ex.size = 0.12
Average Trees (random)	88.95%	78.57%	50.87%	90.13%	9.87%	64.89%	ex.size = 0.45, tree.nr = 200
Decision Trees	96.51%	77.14%	91.67%	99.33%	0.67%	93.08%	probability threshold: 0.5
Random Forests	100%	100%	100%	100%	0%	100%	probability threshold: 0.6
Logit	91.86%	43.57%	85.42%	98.24%	1.76%	60.3%	probability threshold: 0.75
Probit	92.44%	43.57%	93.75%	98.91%	1.09%	65.3%	probability threshold: 0.75
k-NN	91.86%	68.57%	50%	95.7%	4.3%	60%	k = 6, # of principal components: 2
BRT	94.19%	73.57%	88.33%	97.32%	2.68%	74.91%	shrinkage = 0.013, probability threshold: 0.45
SVM	88.37%	86.43%	45.63%	88.18%	11.82%	64.95%	γ=0.0005, C=1.3, sigmoid kernel, probability threshold: 0.15
ANN	90.7%	83.57%	54.76%	91.51%	8.49%	70%	probability threshold: 0.54, # of hidden nodes: 5

Table A.2 Germany

Method	Classification Accuracy	Sensitivity	Precision	Specificity	False Alarm	F_1-Score	Comments
Average Trees (K-fold)	89.88%	53.13%	94.44%	98.41%	1.59%	77.62%	ex.size = 0.13
Average Trees (random)	86.9%	62.5%	80.54%	93.65%	6.35%	71.02%	ex.size=0.225, tree.nr=200
Decision Trees	89.29%	56.25%	94.44%	98.41%	1.59%	80%	probability threshold: 0.8
Random Forests	100%	100%	100%	100%	0%	100%	probability threshold: 0.37
Logit	84.52%	31.25%	100%	100%	0%	88.89%	probability threshold: 0.85
Probit	84.52%	31.25%	100%	100%	0%	88.89%	probability threshold: 0.85
k-NN	97.62%	87.5%	100%	100%	0%	91.67%	k = 3, # of principal components: 3
BRT	92.26%	87.5%	81.11%	92.46%	7.54%	81.65%	shrinkage = 0.01, probability threshold: 0.3
SVM	85.12%	59.38%	56.19%	90.48%	9.52%	60.95%	γ=0.0014, C=2.9, radial kernel, probability threshold: 0.35
ANN	88.1%	84.38%	68.52%	88.29%	11.71%	73.33%	probability threshold: 0.67, # of hidden nodes: 5

Table A.3 Japan

Method	Classification Accuracy	Sensitivity	Precision	Specificity	False Alarm	F_1-Score	Comments
Average Trees (*K*-fold)	93.6%	77.13%	84.32%	92.42%	7.58%	88.61%	ex.size = 0.09
Average Trees (random)	93.6%	77.13%	84.32%	92.42%	7.58%	88.61%	ex.size = 0.1 , tree.nr = 200
Decision Trees	89.53%	39.63%	100%	100%	0%	82.22%	probability threshold: 0.8
Random Forests	100%	100%	100%	100%	0%	100%	probability threshold: 0.45
Logit	87.79%	66.46%	56.23%	87.9%	12.1%	59.06%	probability threshold: 0.35
Probit	87.79%	66.46%	56.23%	87.9%	12.1%	59.06%	probability threshold: 0.35
k-NN	98.84%	95.43%	95.43%	99.21%	0.79%	95.43%	k = 2, # of principal components: 3
BRT	97.67%	95.43%	93.02%	96.3%	3.7%	93.52%	shrinkage = 0.035, probability threshold: 0.3
SVM	87.21%	95.43%	48.56%	81.62%	18.38%	70.73%	γ=0.0049, C=1.2, radial kernel, probability threshold: 0.3
ANN	69.77%	63.41%	31.34%	63.42%	36.58%	55.91%	probability threshold: 0.5, # of hidden nodes: 6

Table A.4 Mexico

Method	Classification Accuracy	Sensitivity	Precision	Specificity	False Alarm	F_1-Score	Comments
Average Trees (*K*-fold)	92.4%	62.5%	73.33%	97.45%	2.55%	83.52%	ex.size = 0.12
Average Trees (random)	92.4%	72.5%	66.67%	96.07%	3.93%	79%	ex.size = 0.1 , tree.nr = 200
Decision Trees	92.4%	62.5%	73.33%	97.45%	2.55%	83.52%	probability threshold: 0.5
Random Forests	100%	100%	100%	100%	0%	100%	probability threshold: 0.6
Logit	86.55%	15%	60%	98.41%	1.59%	66.67%	probability threshold: 0.5
Probit	87.72%	25%	86.67%	98.41%	1.59%	48.89%	probability threshold: 0.4
k-NN	100%	100%	100%	100%	0%	100%	k = 1, # of principal components: 4
BRT	93.57%	70%	92%	98.41%	1.59%	74.03%	shrinkage = 0.01 probability threshold: 0.3
SVM	49.71%	42.5%	14.38%	48.62%	51.38%	32.78%	γ=0.006, C=0.08, radial kernel, probability threshold: 0.16
ANN	85.96%	47.5%	45.24%	92.45%	7.55%	47.87%	probability threshold: 0.55, # of hidden nodes: 3

Table A.5 United Kingdom

Method	Classification Accuracy	Sensitivity	Precision	Specificity	False Alarm	F_1-Score	Comments
Average Trees (K-fold)	89.82%	62.5%	61.11%	93.93%	6.07%	70%	ex.size = 0.25
Average Trees (random)	89.82%	62.5%	61.11%	93.93%	6.07%	70%	ex.size = 0.15, tree.nr = 200
Decision Trees	90.42%	78.13%	51.76%	90.94%	9.06%	68.47%	probability threshold: 0.5
Random Forests	97.6%	86.46%	100%	100%	0%	91.96%	probability threshold: 0.7
Logit	88.02%	20.83%	62.5%	97.13%	2.87%	71.43%	probability threshold: 0.5
Probit	86.83%	12.5%	50%	97.13%	2.87%	50%	probability threshold: 0.5
k-NN	89.82%	50%	48.89%	95.73%	4.27%	84.44%	k = 10, # of principal components: 2
BRT	91.02%	59.38%	75.42%	94.33%	5.67%	59.17%	shrinkage = 0.01 probability threshold: 0.3
SVM	86.83%	0%	0%	100%	0%	0%	γ=0.0003, C=0.009, radial kernel, probability threshold: 0.95
ANN	86.83%	15.63%	50%	95.93%	4.07%	35%	probability threshold: 0.51, # of hidden nodes: 4

Table A.6 United States of America

Method	Classification Accuracy	Sensitivity	Precision	Specificity	False Alarm	F_1-Score	Comments
Average Trees (K-fold)	94.64%	80%	76.67%	96.03%	3.97%	75.81%	ex.size = 0.1
Average Trees (random)	94.64%	80%	76.67%	96.03%	3.97%	75.81%	ex.size = 0.1, tree.nr = 200
Decision Trees	95.24%	80%	79.33%	96.83%	3.17%	77.59%	probability threshold: 0.5
Random Forests	100%	100%	100%	100%	0%	100%	probability threshold: 0.4
Logit	92.26%	40%	93.33%	99.21%	0.79%	71.53%	probability threshold: 0.8
Probit	90.48%	25%	83.33%	99.21%	0.79%	71.43%	probability threshold: 0.92
k-NN	92.26%	55%	77.78%	96.83%	3.17%	81.9%	k = 10, # of principal components: 2
BRT	92.26%	40%	90%	99.21%	0.79%	94.44%	shrinkage = 0.009, probability threshold: 0.5
SVM	91.07%	65%	71.79%	93.65%	6.35%	67.07%	γ=0.00025, C=0.92, radial kernel, probability threshold: 0.25
ANN	89.29%	65%	53.73%	91.27%	8.73%	62.93%	probability threshold: 0.52, # of hidden nodes: 5

References

Aggarwal, C. C. (Ed.). (2015). Data Classification - Algorithms and Applications. Boca Raton, FL (USA): CRC Press, Taylor & Francis Group.

Baltagi, B. H. (2002). Econometrics (3rd ed.). Berlin; Heidelberg (Germany): Springer-Verlag. doi:10.1007/978-3-662-04693-7

Barnett, W., Serletis, A., & Serletis, D. (2015). Nonlinear and Complex Dynamics in Economics. Macroeconomic Dynamics, 19(8), pp. 1749-1779. doi:10.1017/S1365100514000091

Bishop, C. M. (2006). Pattern Recognition and Machine Learning. New York, NY (USA): Springer Science and Business Media.

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

Christiansen, C. (2013). Predicting severe simultaneous recessions using yield spreads as leading indicators. Journal of International Money and Finance, 32, pp. 1032-1043. doi:10.1016/j.jimonfin.2012.08.005

Claessens, S., & Kose, A. M. (2009, Mar. IMF Publications,). What Is a Recession? Finance & Development, 46(1), pp. 52-53. Retrieved from http://www.imf.org/external/pubs/ft/fandd/2009/03/pdf/basics.pdf

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

Dovern, J., & Huber, F. (2015). Global prediction of recessions. Economics Letters, 133, pp. 81-84. doi:10.1016/j.econlet.2015.05.022

ECRI. (2013). Business cycle peak and trough dates, 1948–2011. Retrieved (by the authors) Sep. 19, 2013, from https://www.businesscycle.com/ecri-business-cycles/international-business-cycle-dates-chronologies

Estrella, A., & Mishkin, F. S. (1998, Feb.). Predicting U.S. Recessions: Financial Variables as Leading Indicators. The Review of Economics and Statistics, 80(1), pp. 45-61. doi:10.1162/003465398557320

Fisher, I. (1933, Oct.). The Debt-Deflation Theory of Great Depressions. Econometrica, 1(4), pp. 337-357. doi:10.2307/1907327

Friedman, M. (1968, Mar.). The Role of Monetary Policy. The American Economic Review, 58(1), pp. 1-17. Retrieved Apr. 2, 2018, from http://www.jstor.org/stable/1831652

Friedman, M., & Schwartz, A. J. (1963). A Monetary History of the United States, 1867-1960. Princeton, NJ (USA): Princeton University Press.

Gogas, P., Papadimitriou, T., Matthaiou, M., & Chrysanthidou, E. (2015, Apr.). Yield Curve and Recession Forecasting in a Machine Learning Framework. Computational Economics, 45(4), pp. 635-645. doi:10.1007/s10614-014-9432-0

Goodwin, R. M. (1951, Jan.). The Nonlinear Accelerator and the Persistence of Business Cycles. Econometrica, 19(1), pp. 1-17. doi:10.2307/1907905

Hall, R. E., & Lieberman, M. (2013). Macroeconomics: Principles & Applications (6th ed.). Mason, OH (USA): South-Western, Cengage Learning.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An Introduction to Statistical Learning - with Applications in R. New York, NY (USA): Springer Science and Business Media. doi:10.1007/978-1-4614-7138-7

Kauppi, H., & Saikkonen, P. (2008, Nov.). Predicting U.S. Recessions with Dynamic Binary Response Models. The Review of Economics and Statistics, 90(4), pp. 777-791. doi:10.1162/rest.90.4.777

Keynes, J. M. (1936 [2013]). The Collected Writings of John Maynard Keynes: The General Theory of Employment, Interest and Money (4th ed., Vol. VII). New York, NY (USA): Cambridge University Press.

Kiani, K. (2008). On Forecasting Recessions via Neural Nets. Economics Bulletin, 3(13), pp. 1-15. Retrieved May 27, 2018, from https://pdfs.semanticscholar.org/af11/30f6a52e1e41057d001a365f78968a2bbfe7.pdf

Knoop, T. A. (2015). Business Cycle Economics: Understanding Recessions and Depressions from Boom to Bust. Santa Barbara, CA (USA): ABC-CLIO, LLC.

Kubat, M. (2017). An Introduction to Machine Learning (2nd ed.). Cham, ZG (Switzerland): Springer International Publishing AG. doi:10.1007/978-3-319-63913-0

Marx, K. (1894 [2010]). Capital: A Critique of Political Economy (Vol. III). (F. Engels, Ed.) New York, NY (USA): International Publishers.

NBER. (2010, Sep. 20). US Business Cycle Expansions and Contractions. Retrieved May 25, 2018, from http://www.nber.org/cycles/

OECD. (2018a). "Business confidence index (BCI)" (indicator). Retrieved Mar. 27, 2018, from http://dx.doi.org/10.1787/3092dc4f-en

OECD. (2018b). "Long-term interest rates" (indicator). Retrieved Apr. 6, 2018, from http://dx.doi.org/10.1787/662d712c-en

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

Ricardo, D. (1821 [2001]). On the Principles of Political Economy and Taxation (3rd ed.). (reprint), Kitchener, Ontario (Canada): Batoche Books.

Schumpeter, J. A. (1942 [1994]). Capitalism, Socialism and Democracy (5th ed.). London (UK): Routledge.

Smith, A. (1776 [1977]). An Inquiry into the Nature and Causes of the Wealth of Nations (5th ed.). (E. Cannan, Ed.) (reprint), Chicago, IL (USA): University Of Chicago Press.

Tsoulfidis, L. (2010). Competing Schools of Economic Thought. Berlin; Heidelberg (Germany): Springer-Verlag. doi:10.1007/978-3-540-92693-1

Wong, A. Y.-T., Pak, T., & Fong, W. (2011, Dec.). Analysing interconnectivity among economies. Emerging Markets Review, 12(4), pp. 432-442. doi:10.1016/j.ememar.2011.06.004