LUISS

Department of

Business and Management

Bachelor's Degree in

Management and Computer Science

Course of Data Analysis for Business

MACROECONOMICS FORECASTING USING MACHINE LEARNING: AN APPLICATION TO ITALIAN INDUSTRIAL PRODUCTION

Prof. Francesco Iafrate

SUPERVISOR

Leonardo Berrettoni-249781

CANDIDATE

Academic Year 2021/2022

To my grandfather Lido that has been always the first one to cheer for my university achievements and would have been very happy and proud of seeing his grandchild graduated.

Index

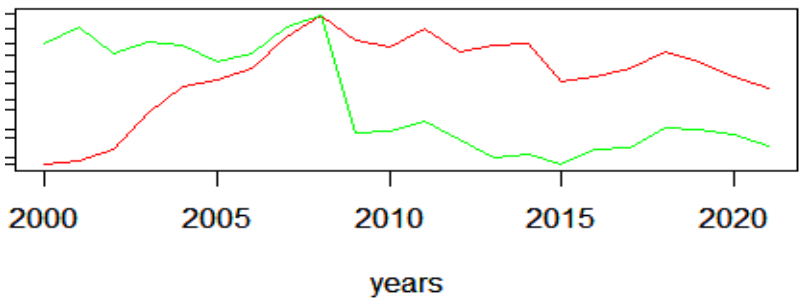
INTRODUCTION

"We are facing a recession period", "The GDP has grown by 3 percentages points", these and many other different sentences are expression that all of us has become used to. They are used when someone is talking about the GDP. But what does GDP stand for? And what is it?

GDP stands for Gross Domestic Product. It measures the monetary value of all the goods and services produced by a country in a given period of time or, using other words, it measures the wealth of a country. Macroeconomically speaking it is composed by the sum of different factors: GDP=C+I+G+(X-IM) where C is the consumption, I is the investment, G is the expenses of the government and X-IM is the net export (exports-imports).

GDP depends on different factors but one of the most influent is the Industrial Production. As OECD says, the industrial production of a country measures the output of the industrial sector that comprises mining, manufacturing, electricity, gas and steam and air-conditioning. This indicator is the most important when someone talks about GDP because it is the one that can catch in the best way the change in consumer's behaviour and the demand. This statement can be observed also looking at figure (I) where are depicted, scaled taking 2000 as base year, the trends on the industrial production (the green curve) and of the GDP (the red curve) of Italy during the last 21 years. What can be noticed is how, even if with different intensity, the two curves move up and down nearly in the same way; this can be seen very clearly from 2005 onwards, where the two curves reach the peaks and the drops almost at the same period of time.

Figure (I)



GDP and Industrial Production represent a sign health of an economy because, usually speaking, when they rise it means that the wages of the people working in that specific country are rising too, that the unemployment rate is decreasing and this situation facilitates and promotes investment scenarios and this brings prosperity to a country's economy; when the two indicators go down, having a negative delta, instead, it means all the opposite, so unemployment rate goes up, wages go down and a ring of negativity stirs in a country's economy.

Understanding the behaviour and the values of the GDP and of the Industrial Production, therefore, is important to understand the state of health of country and so to decide which policies a State might undertake to better help its citizens.

As of now all seems to be clear and straightforward, it's just enough to read some data and then take decisions accordingly. In the real-world life is not so easy. GDP is released quarterly (each three months) while some of the other indicators that compose the GDP are released monthly. The problem arises when we look at the time when they are released. GDP data are released about 8 weeks after the end of the reference quarter and some data about the other indicators are released with some delay. In this scenario becomes of paramount importance the forecast of this variables. Luckily for us, we are all leaving in a world where the technology is developing at a very fast pace and this technological evolution enabled us with some very useful machine learning algorithms that can help us solving our problem. In our specific case we will use these machine learning methods through the help of an environment called RStudio. During the research will be used supervised (Elastic net and XGBoost) and autoregressive (SARIMA) methods. Supervised methods are methods that must be trained before being applied to a new set of data. So, when these types of methods are used, there is the need for a training set, where the model can be trained finding the best parameters according to the processed data, and a test set, where the performances of the algorithm can be tested. Both the training set and the test set for consistency reasons come from the same initial dataset; the latter, indeed, is divided before starting to construct the models.

Autoregressive models, instead, work a bit differently. They are mostly used when someone has to handle time series, they use previous values of the response variable as a regressor for itself. In this case the seasonal autoregressive model (SARIMA) that will be used as a benchmark for the valuation of the other models: if

the other models perform better than the SARIMA, then they will be reliable models, otherwise if they perform worse, they will not be good forecasting models. As said before our aim is to face and solve somehow the problem that the lag that the release of GDP data creates. This paper though, will focus on a step before the GDP; indeed, machine learning methods will serve the aim to forecast the value of the industrial production that, as highlighted previously, is the most important GDP's indicator. But however, so far, when dealing with industrial production forecasting are not always used machine learning methods but, instead, traditional econometric methods.

SOGEI (the Information and Communication Technology Company of the Ministry of the Economy and Finance.), a firm with whom I collaborated for putting into writing my thesis and that operates in the ICT sector, and it is totally owned by the Italian MEF (ministry of economic and finance), indeed, when dealing with the forecasting of the Italian industrial production uses the following method: it has a model that is divided in two different parts, one that takes into consideration the energetic component (that has a weight of around 10% in the overall index) and another component that takes into consideration all the other components but the energetic one. Both components are forecasted using the OLS (ordinary least square) method that includes monthly economic indicators that are realised with higher promptness than the industrial production. These indicators are divided into quantitative ones (also called hard) like for example the highway traffic of heavy vehicles, the production of electric energy, the gas consumption of the energy company, and into qualitative ones (called also soft because they express judgement and expectations) as for example PMI and ISTAT climate of trust. When these indicators are not available, they are forecasted using an ARIMA model. At the end, a forecast of the two initial components is obtained, and after having weighted them, they obtain the value of the industrial production.

The response variable of the model is the Industrial Production of Italy (indicated in the dataset as ip_target), a continuous variable. All the R code is focused on analysing this variable, understand which variables are the most relevant in forecasting the Italian industrial production and try to construct the best possible model to forecast its future values before official data are released. Since, as said previously, the response variable is continuous, in order to achieve the goal just listed, the problem faced will be a "regression" problem. The methods used are the ones mentioned before

and that will be better explained in a sequent section in which we will go deeper in the mechanism theoretically and mathematically.

OBJECTIVE AND SCOPE

So, to wrap up the questions that we are going to try to answer during this paper are:

(i) Which is the most efficient way to estimate the Italian industrial production?

(ii) which is the best algorithm or method to use? How much the forecast is accurate?

These questions represent the fil rouge of all the study and will try to be answered in the next sections.

1. DATASET

The first thing to do in order to reach the pre-set goal is assessing the dataset. It has been gently given by SOGEI. The dataset I was supplied with, the one usually used by the company, was a quite big one: the most important variable was the Italian industrial production (ip_target), an historical series whose data were available starting from January 1970 to December 2021 (636 observations). In addition to the "target variable", that represents the response variable, the dataset was composed by other 317 variables regarding the confidence in the manufactural sector in different countries, economic sentimental indicator, exports, exchange rates, unemployment rate, industrial production of different countries and many other.

What can be immediately noticed, looking at the dataset, is that it is not complete, indeed it can be said to be a sparse dataset. This can be explained by the fact that the recording of some variables has been interrupted before December 2021 or they have just different updating times, while some others started to be recorded after January 1970. Moreover, there are some missing data also in between some variables and this can be caused by a missing in the registration of the data.

All these missing values could have represented a problem for the implementation of the algorithms, because many machine learning methods need to work with complete datasets or are able to handle only a small number of missing values. Once understood the criticalities that a sparse dataset could have created i was put in front to a decision: how can I deal with these missing values? Discussing the issue also with my tutors in SOGEI we have decided to take a decision that might seem drastic, but it was the only reasonable one: we erased some of the rows and some of the columns of the dataset. The criterions we used were two: (i) first of all we decided to erase all the rows containing data regarding all the years before 2000 and after 2021. We did this because before 2000 the Euro as official currency was far to be introduced and the data regarding 2022 were data up until February so not few to be considered relevant; then (ii) we decided to erase all the columns in which recorded data started with a delay of more than 6 months (values were missing from above) and variables where data stopped to be recorded more than 6 months in advance. This procedure was necessary for two reasons: the first one is because otherwise neither imputation functions in R would have been able to provide for so many missing values and the

second one is that even if functions had filled the empty cells the reliability of those results would not have been enough.

After this work on the initial dataset, there were no more available some variables as the number of registered cars that are not Fiat, Lamborghini, Lancia, Maserati, Ferrari or Alfa Romeo, the deflation of retail sales and unemployed people. Once performed these operations we ended up with a dataset that was smaller than the initial one; its dimensions, indeed, are 265 rows and 318 variables. However, the most important gain was the fact that the final dataset had many missing values less. Some values were still missing but this was not a big deal because there were few and they could have been tried to be imputed with a function in R called "*mice".*

The dataset that will came out with after the previous mentioned adjustments, and at the end of the pre-processing phase, will be the working dataset on which all the analysis will be based.

A first look to the working dataset highlights how, despite the *id* variable and the time-related variables, all the columns of it are composed by numbers. This means that all over the R-code only continuous variables will be faced, and for this reason there will not be the need to handle continuous and categorical variables at the same time, making our life much easier because we will have only one unique type of variables and we can handle them all in the same way

Let's dive into the R-code, starting with a preliminary analysis of the response variable.

Table (I)

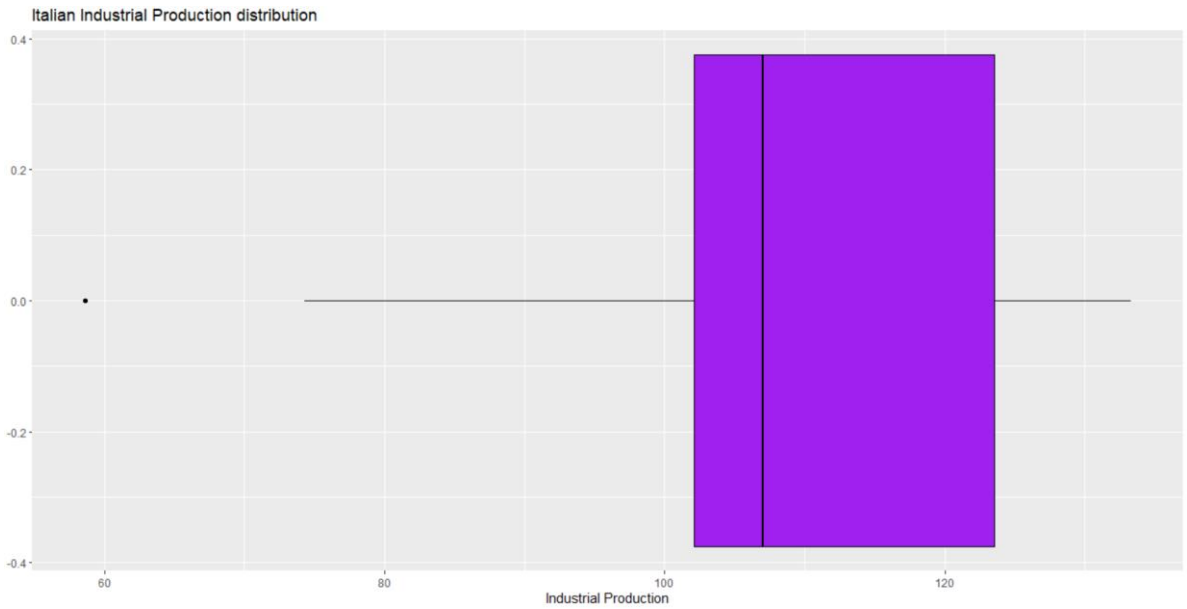
Min	1st Qu.	Median	Mean	3rd Qu.	Max
58,6	102,1	107	111,6	123,5	133,3

From table (I) we can have a look to the range of the response variable that goes from 58.6 to 133.3. Moreover, we can observe that the mean value of the industrial production in 111.6, a value that takes into consideration an outlier and the values of the two big drops that happened and that will be seen more deeply later. The standard deviation, instead, is equal to

12 (this shows that are not too much spread out but varies around the 10% from the mean. This value is also influenced by the outlier previously mentioned).

A useful graph can be the one showing the quartiles of the monthly distribution of the industrial production like the following boxplot.

Figure (II)

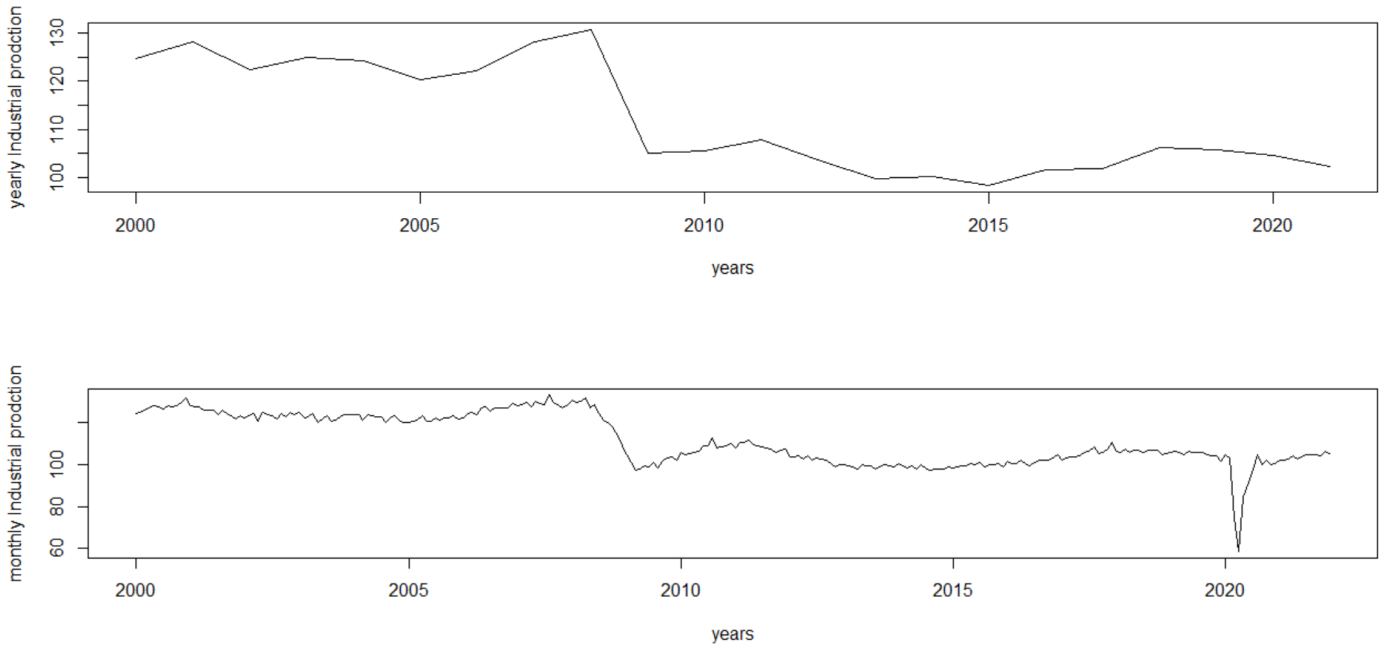


From figure (II) can be seen clearly how the response variable is distributed, which is its min, its max and its quartiles. We can state that it is not a symmetric distribution, indeed the dimensions of the 4 quartiles are all different between themselves showing that the observations are more concentrated in some quartiles (like the second and the fourth) and more spread in the others like the first and the third).

Moreover, it is possible to notice a black dot in the left part of the figure. It is an outlier. An outlier is an observation that lies in an atypical position, too much distanced from the other ones. That black dot corresponds to the value 58.6 that is the minimum value that our response variable assumes. This is considered an outlier because the distance between it and the next value (74.3) is way bigger than other distance between two consecutive observations (if we consider the sorted distribution of industrial production's values).

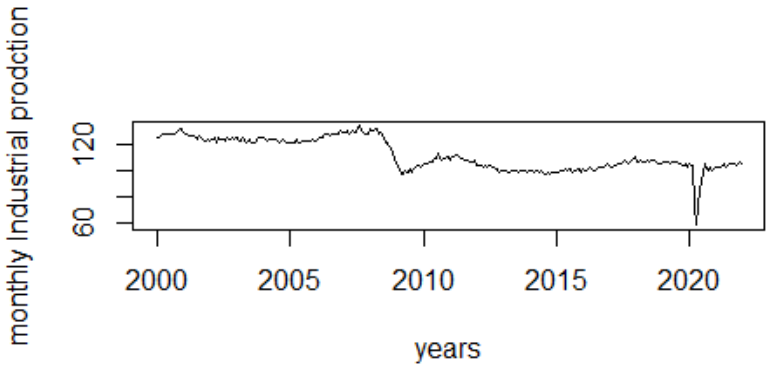
So far, analysing the quartiles and the boxplot, we have considered the observation of the response variable to be sorted; but it is important also to take a look at the trend of the depended variable in the way as it is. There will both be shown the yearly and the monthly movements of the Italian industrial production in the past 20 years.

Figure (III)



On the upper side of the figure, it is shown the yearly evolution of the response variable, while on the downer side the monthly one. Obviously, the behaviour of the two plots is almost the same because they represent the same variable, but it is clear how the downer graph is able to catch also smaller changes in the values of our response variable. A deeper look of the two graphs also shows another difference. In the upper graph it is possible to notice one big drop in the industrial production of the country that corresponds to the 2007-2008 financial crisis, while on the downer graph it is possible to notice two big drops.

Figure (IV)



Looking more in dept at the monthly graph, it is clear how the second big drop happens just after the start of 2020, so it corresponds to the start of the spread of the COVID-19 pandemic where the industrial production reaches, by far, the minimum value ever in the recent years. Probably this big drop it has not been caught by the yearly graph because after that big fall there is a likewise rise of the industrial production, and since all this happens in some months, but within the same year, the yearly graph is not able to catch this movement.

Another insight that we can grasp from the figure is that, contrary to the COVID-19 related drop, after the fall in the value caused by the 2007 crisis the value of the industrial production has never come back to the pre-crisis values. We can see, indeed, that after the drop the graph goes up for a while but without reaching the past values.

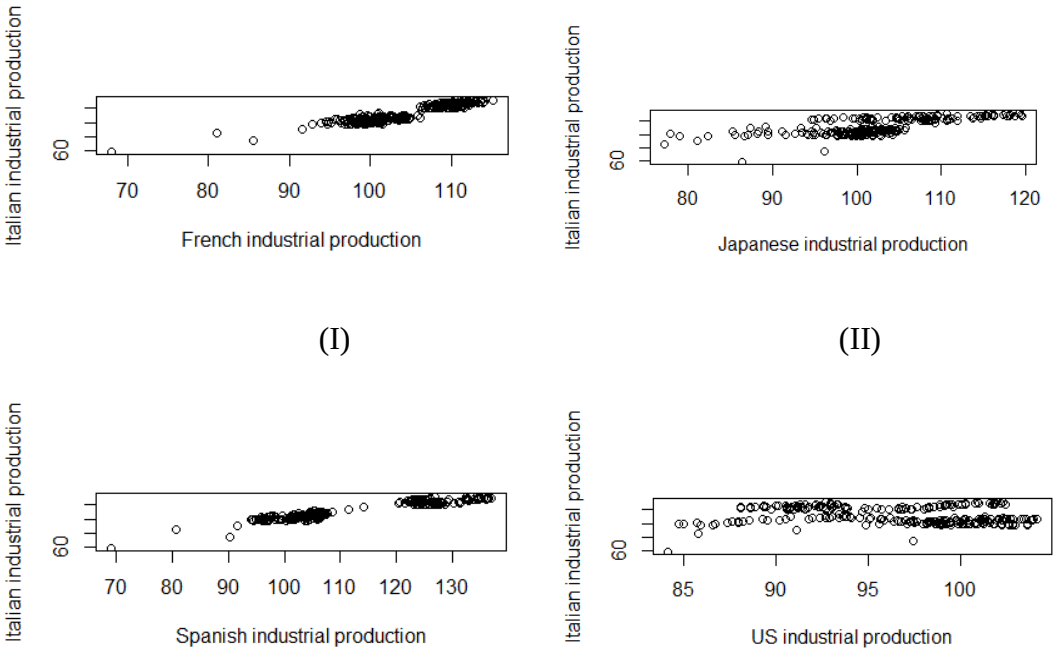
1.1 Significant relations

To end this first introductory part about the dataset and the response variable, we should ask ourselves whether the Italian industrial production is significantly related to other variables of the dataset. It makes sense to compare the Italian industrial production with the one of other countries; we will compare it with the industrial production of 4 countries, 2 within the UE (Spain and France) and 2 outside the UE (US and Japan). In order to understand whether a significant relation between the Italian industrial production and these variables is present or not, we will use two methods: a graphic method, printing a scatter plot of the values of the two industrial production at the same period of time and looking whether a sort of path can be found, and a more theoretical and rigorous one, creating a linear model and looking at the p-value. P-value is one the output of the linear function, we want it to be the smaller as possible because a p-value small enough means that we can be confident that the coefficient of the variable analysed is not zero, meaning that the variable adds value to the model. P-value indicates the probability that your data could have occurred under the null hypothesis (that in our case is "the variables is not significant in the forecasting of the Italian industrial production"). The threshold that will be used for the p-value is 5%, or, in other words, we want it to be less than 5% to reject the null hypothesis.

1.1.1 Graphical Method

First let's focus on the graphs:

Figure (V)



(I) (II)

(III) (IV)

Focusing on the sub-graphs I and III it is possible to notice that the Italian industrial production seems to have a significant linear relation with the Spanish and French one, since the behaviour of the points printed in the graphs can be explained using a line with a positive slope. Sort of the same reasoning can be made for the relation with the Japanese industrial production, even if less evident than the previous two cases also in this case can be glimpsed a linear relation between the two variables. A different statement can be made for what concerns the graph IV where the dots seem to shape an X; but this doesn't exclude the possibility of a significant linear relation, we will examine better the case using the second method.

1.1.2 Linear Model Method

As explained above a more rigorous method is to create a linear model to check if a variable is significant for the forecasting of the Italian industrial production. 4 models have been created and the results are the following.

Table (II)

Variable	Coefficient estimate	Pr(>|t|)	Adjusted R-squared
Japanese I.P.	1,109	<2e-16	0,4736
French I.P.	1,833	<2e-16	0,8734
Spanish I.P.	0,863	<2e-16	0,9252
US I.P.	-0,448	0,00316	0,03278

Looking at the results of the 4 models it is possible to notice how the coefficients of the Japanese, French and Spanish industrial production are all positive while the i.p. of the US has a slightly negative linear relationship with the Italian one. Moreover, we can state that all 4 variables are significant since all the p-values are smaller than 5%.

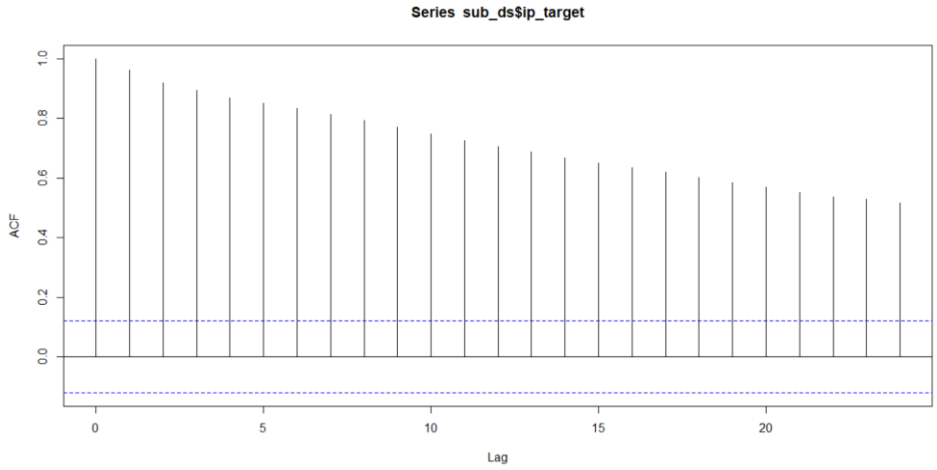
The considerations made looking just at the graphs are confirmed also with this second method because, looking at the adjusted R-squared, that explains how much each variable explains the variation within the Italian industrial production, we can notice that the variables that have the highest value are the French and the Spanish i.p., that were the variables that, graphically, had the clearest linear relation with the Italian industrial production. Same argument can be said for the US industrial production that is the variables that graphically was the least linearly related to our response variable and also in this case is the one that has the lowest adjusted R-squared and the highest p-value, even if it remains significant.

1.2 Stationarity

Before starting to construct the different models since we are dealing with time series data, their stationarity must be checked. Data are said to be stationary if their properties does not depend on the time in which data have been observed. Usually, data that have trends or seasonality are not stationary. Since our data seem to be seasonal data, it is worth to take a deeper look to their stationarity.

There are two ways to understand if data are stationary or not: the first one is to take a look at the ACG graph (the one printed below), if the ACF drops to zero relatively quickly data are stationary otherwise they are not; the second method is to perform the unit root test. Unit root test is statistical hypothesis testing is applied in order to check stationarity; in our case the test applied is the Kwiatkowski-Phillips-Schmidt-Shin (KPSS). In this test the null hypothesis is that "data are stationary"; small p-values (e.g., less than 0.05) suggest that we can reject the null hypothesis and so differencing is required. We have to check it focusing on the value of the test statistic. (Prabhakaran, 2022)

Figure (IX)



From the ACG graph we can immediately notice how the data are not stationary. Indeed, the ACF value does not even drop to zero and it decreases very slowly.

What was an initial insight looking at the ACF graph is confirmed by the KPSS root test. Indeed, looking at the value of the test-statistic, it is possible to notice how this value is much bigger that the critical value of both 1% and 5% significance level;

this means that the null hypothesis, “the data are stationary”, can be rejected and that the data are non-stationary.

Table (III)

Value of test statistic
3,2863

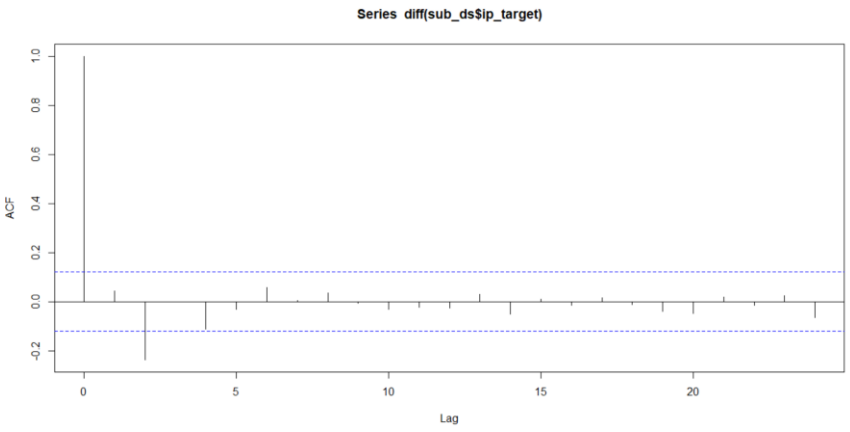
Table (IV)

	10%	5%	2,5%	1%
Critical Values	0,347	0,463	0,574	0,739

Now that is clear that our data are not stationary, what do we have to do? The answer to this question is Differencing. Differencing, as explained in the research of Hyndman and Athanasopoulos (2018), is just computing the difference between two consecutive observations.

After differencing the data, the two previously explained methods have been applied again.

Figure (X)



Looking at the ACF graph after differencing, can be immediately noticed that the path of the graph is different. Even if there are still some lags that are outside the blue dotted lines, all the others are inside the range shown by those lines and also ACF drops to zero quite quickly.

If the graph can result not so reliable, any possible doubt disappears when KPSS is applied again:

Table (V)

Value of test statistic
3,2863

Table (VI)

	10%	5%	2,5%	1%
Critical Values for significance levels	0,347	0,463	0,574	0,739

We can notice how after differencing the value of the test-statistic is much smaller than the critical values of the 1% and 5% significance values meaning that now the data are stationary.

2. PREPROCESSING

After taking a look to how the response variable is composed and how it behaves over time, it is proper to take a step back and talking again about the dataset. As it was said in previous section the dataset is composed by 318 columns (variables). One fast way to check if there are some variables that are redundant is checking the correlation between them. Correlations explains the relationship between variables, using a range between -1 and 1: a correlation equal to 1 means that the variables move exactly in the same way, correlation equal to -1 means that the variables move in a completely opposite way. In the construction of a forecasting model correlation equal to 1 (or close to 1) is not seen positively: it means, indeed, that the two perfectly correlated variables explain the same portion of the dataset and therefore keeping both of them is redundant. This make sense because when a forecasting model must be created, people want it to be as simple as possible so it must have high performances but having the smallest number of variables as possible. For this reason, we will first check correlation among all the variables (variable vs. variable) to see whether some variables are highly correlated and so interchangeable in the creation of the forecasting model. This method of comparing each variable against each other variable is useful to better understand the dataset while the real correlation method used to perform variable selection is the one computing correlation of each variable against the response variable, looking for those ones that mostly explain the behaviour of the dependent variable (2.2.1).

To check correlations in R is very simple: function “cor” comes to help us. R offers different ways also to visualize graphically how much variables are correlated between themselves but in this case, since the number of variables to analyse is too big, we will need to settle for a simple list of correlations.

The result is a very long list of correlations where some of them have very high correlation (equal to 1 or very close to 1). This high values of corelation can be explained by the fact that some variables of the dataset explain the same thing but under different points of view or some others are just a subset of some variables and so the result is that they are highly correlated. Considering 0.75 as the highest safe value for correlations, almost one-third of the table is filled with too big correlations.

This suggest that there are a lot of redundant variables that can be considered not relevant for the creation of our model.

2.1 Dealing With Missing Data

Before eliminating some variables maybe someone could object that maybe the fact that there still missing values inside our dataset could have an impact on the correlation between variables. R offers us a useful function called “mice”. According to what is written in the R documentation (Mice Function - RDocumentation, n.d.), the function mice is a method that deals with missing data and that “can impute mixes of continuous, binary, unordered categorical and ordered categorical data. In addition, MICE can impute continuous two-level data, and maintain consistency between imputations by means of passive imputation”. In this specific case, as suggested by the R program, the method “cart” has been used within the function. This method uses classification and regression trees in order to do imputation and to fill the missing spots in the dataset.

Despite the use of this very powerful tool there were still some variables that presented some missing values. Since neither the strength of MICE was able to fill completely those variables it has been decided to eliminate them from the working dataset. So, the new working dataset was composed by all the variables as before but the variables that still contain missing values (Mostly variables related to the consumer confidence or to manufacturing factories confidence).

After this correlation between variables was computed again and despite the imputation there were still too many high correlations. This makes us sure that not all the variables are relevant for the model but only some are. Some specific methods, like Elastic net and eXtreme Gradient Boosting, will be used to find out the best subset of variables.

2.2 Feature Selection

2.2.1 Correlation Method

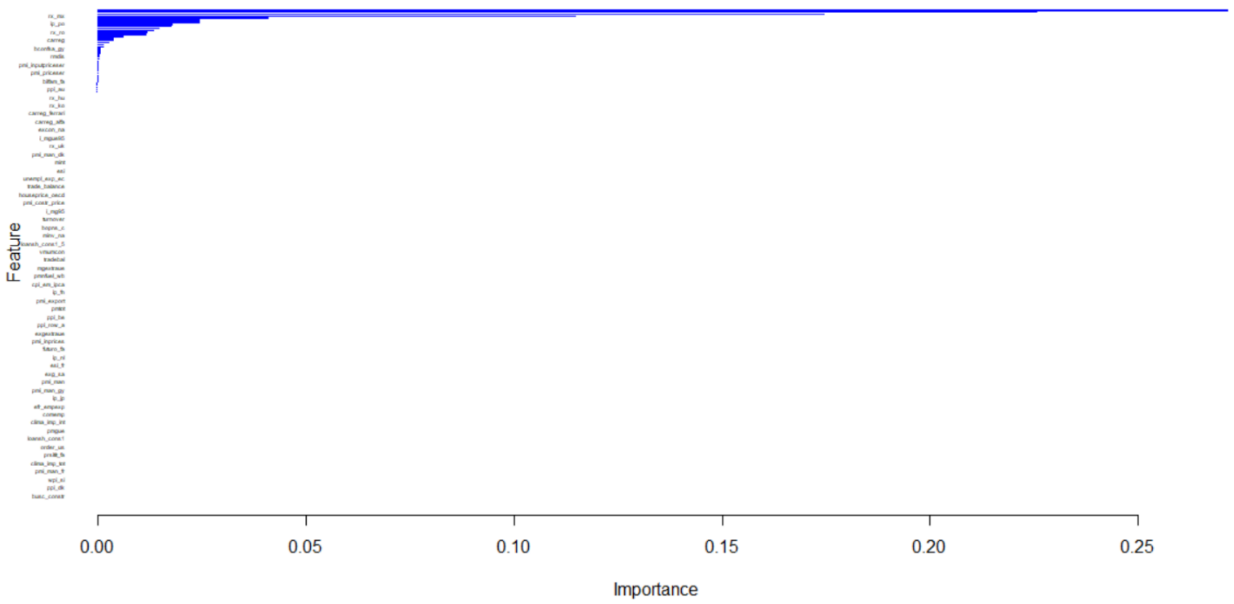
With a simple line of code in R we can see how much each variable of the actual dataset is correlated to the dependent variable. Since we are looking for the most correlated variables, we have to sort them. In order to do this, excel will help us. The output of the previous line of code has been exported in excel and through excel the correlations have been sorted and then uploaded into R again. But clearly what we have to do is performing variable selection so not all the variables will be taken. We are looking for the variables that have the correlation value, taken in absolute value, bigger than 0.5. We have taken the absolute value because we are interested in the variables that are positively correlated with the response variable, and therefore sort of same behaviour; but we are also interested in those variables that are highly negatively correlated with the response variable because they are useful to explain its behaviour too, since they move in an opposite way with respect to the dependent variable. The variables that turned out to satisfy this criterion have been 151.

2.2.2 XGB Method

The second method used is based on an internal feature of the function XGBoost. What must be performed is to create an extreme gradient boosting model and then access the variable importance coming from the trained model. Also in this case we will need a criterion for choosing the most relevant variables: indeed, we are looking for those variables that are able to cover at least the 95-97% of the dataset variability.

The variables importance plot coming out from the XGB model is the following.

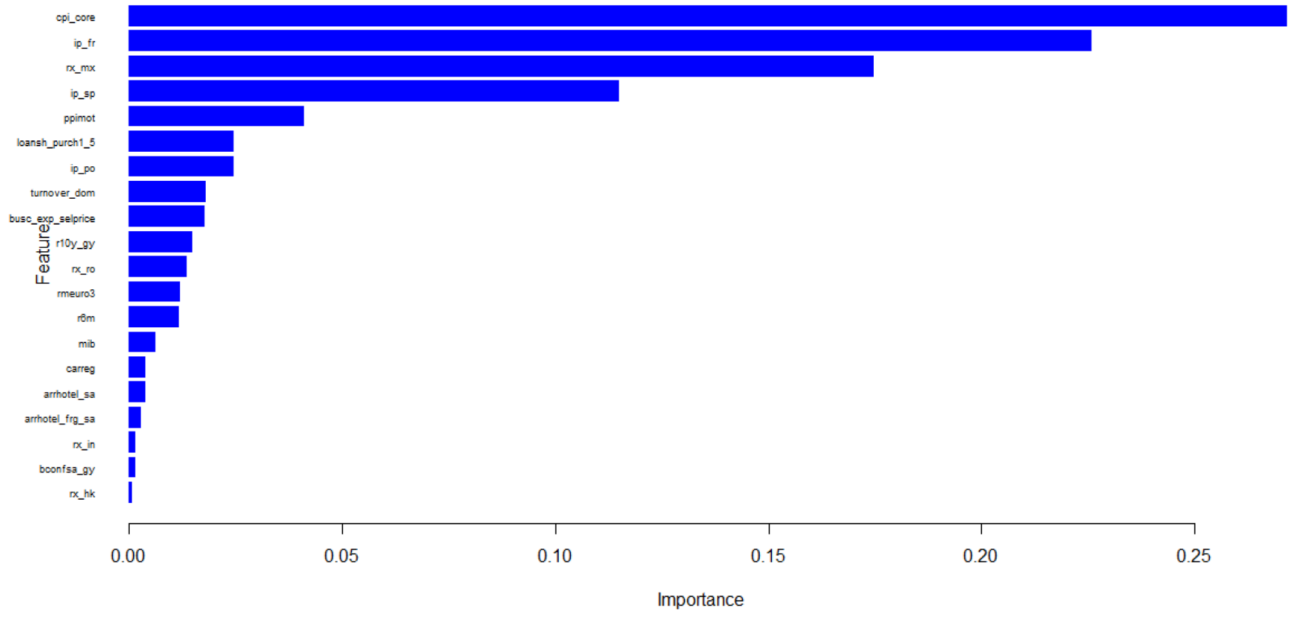
Figure (VI)



From this figure it is not easy to understand which are the variables that contribute the most in the forecast of the industrial production but what come to the eyes is that the relevant variables are not too much.

Then to better understand which variables are the most relevant, let us zoom the figure in.

Figure (VII)



In this case only the first 20 most important variables are displayed. It is evident how the first 4 variables (consumer price index excluding energy and fresh food(*cpi_core)*, the industrial production of France(*ip_fr)*, the exchange rate between the Mexican peso and the US dollar *(rx_mx)* and the industrial production of Spain *(ip_sp)*) are really the most important according to figure (VII). But do they cover the 95-97% of the dataset variability? Actually, the answer is no. In order to reach the threshold, we need 15 variables.

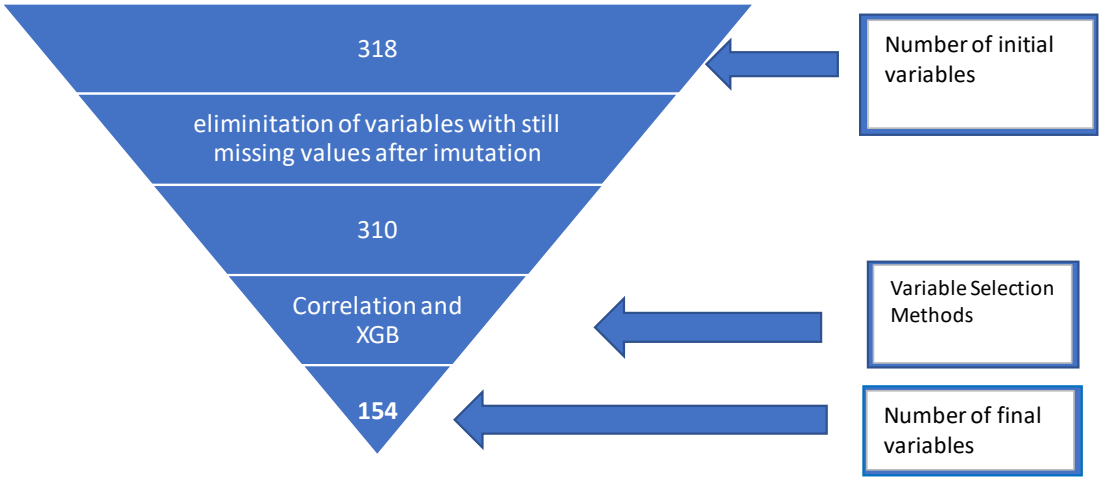
2.3 New Sub Dataset

Once performed the variable selection throughout the help of the correlation and XGB method we need to construct the new dataset. Since we need to take into consideration the results of both methods, in order to select the most important variables, we need to merge the results of the previous explained method. The resulting variables are 153, that can be divided in 11 macro-areas: monetary aggregates, confidence in different aspects (like consumer, or third sectors factories or manufacturing factories), exports, matriculation of different Italian-branded cars, imports, PMI, prices when gods are produced, price when goods are exported, industrial production of different countries, salaries, exchange rates between currencies.

One last step must be performed to conclude the creation of the working dataset. Since our response variable is a time series, time and seasonality of data must be taken into consideration. There are different ways to consider seasonality when dealing with time series. The chosen one in this case was to consider the value of the dependent variable at time t-1 as a predictor: a new variable called *previous_ip_target* is created and in this variable each spot in filled with the value of the industrial production referring to the previous year. In order to do so we downloaded from R an excel file with the dataset containing only the variables selected with the correlation and XGB method, then we manually added the values of the Italian industrial production shifted one step below. After this the new dataset has been uploaded again. To double check the validity of the previous method we checked again the correlations between all the variables of "sub_ds" and the response variable and we noticed that the new variable (the one taking into consideration the values of the response variable at time t-1) was the most correlated with the dependent variable.

Finally, we have our working dataset called *sub_ds*, whose dimensions are 264 observations and 154 variables, and we are ready to construct the forecasting models that will be better discussed in the next section.

Figure (VIII)



3. FORECASTING MODEL

So far, we have been focused on understanding which variables were more relevant in order to better forecast our response variable, the Italian industrial production, and we have been able to construct a sub-dataset containing only a part of all the variables that were present in the initial dataset. What we will focus on from now on, it is to find out the model that will be able to estimate future values of the industrial production making the smallest possible error. Our goal is to perform quarterly prediction of the depended variable (since when dealing with GDP and industrial production it is habit to talk quarterly analysis) using machine learning methods and also an autoregressive method that acts as a benchmark for the valuation of the AI models. The two techniques that will be used to evaluate the models, are the root mean square error (better known as RMSE) and the mean absolute percentage error (better known as MAPE), that will be calculated using the following formulas.

$$\text{MAPE} = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right|$$

$$\text{RMSE} = \sqrt{\frac{\sum_{t=1}^{n}(F_t - A_t)^2}{n}}$$

Where :

n=number of observations

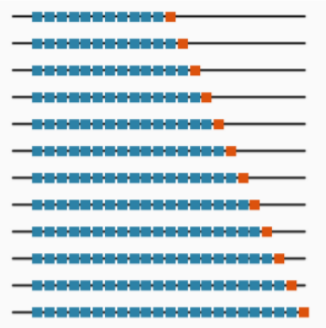
F_t = forecasted value

A_t= Actual value

3.1 Rolling Cross-Validation

When dealing with supervised machine learning algorithms, as the one we will use in this paper, the working dataset must be divided into training set (in order to train correctly the model) and test set (in order to test the accuracy of the created model). In this case this simple method it is not the most correct one. Indeed, when the data are time series one should use rolling cross validation. This method is represented in the figure below (Forecast Evaluation for Data Scientists: Common Pitfalls and Best Practices, 2022).

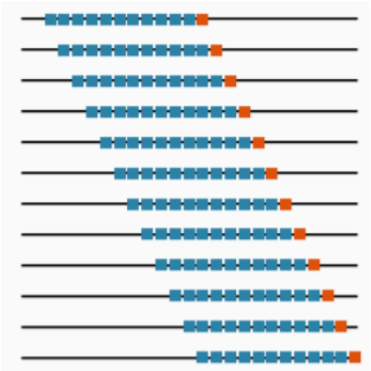
Figure (XI)



Each time the dataset is divided in training set (blue dots) and test set (red dots). But here what changes is that the usual mechanism to evaluate the performances of the model is repeated multiple times. Each time the training part rolls down by one element and thus does the test set and at each iteration the error is calculated. The final error is computed averaging the errors recorded at each iteration.

In the analysed case we have used a special type of rolling cross-validation that can be seen in the figure below.

Figure (XII)



This time the size of the training set is fixed and this window rolls down without changing the size.

In writing down our code, the size of the rolling window has been set to 12 and we have added also another small difference with respect to the two previous images. From the pictures we can see how the test set is represented by the immediate next dot after the end of the training set, but, instead, we have decided to make the test set be the third dot after the end of the training set since experts, when committed in forecasting the industrial production of a country, are used to consider the forecast at a time t+3.

3.2 Seasonal Autoregressive Integrated Moving Average (SARIMA) Model

The first model to be implemented is the SARIMA one, a short way to say seasonal ARIMA; this model therefore is an ARIMA model applied to seasonal data. SARIMA, as well as ARIMA, is an autoregressive model; it means that the response variable forecasting is a linear combination of its own past values; the term autoregression indicates that it is a regression of the variable against itself (Hyndman & Athanasopoulos, 2018). Seasonal ARIMA, besides the non-seasonal parameters (p,d,q) (where p is the lag order so the number of lag observations that are present in the model, d is the degree of differencing so the number of times observations are differenced and q is the size of the moving average window), has also some seasonal parameters $(P,D,Q)_m$ (where P,D and Q means the same as the ones of non-seasonal parameters and m is the number of observations per year).

In our case, in order to create a SARIMA model, we have used a function called auto.arima that automatically compute the model and also all the parameters explained above.

After constructing the model and performing the rolling cross-validation we have to evaluate the performances of the model calculating the RMSE and the MAPE.

3.3 eXtreme Gradient Boosting

The second forecasting method used is XGBoost. It is a short way to say extreme gradient boosting and it is a three based method. The algorithm, as explained by James et al. (2021b) in their book “An introduction to statistical learning”, create in a

sequential way the different trees so that each tree is grown using information coming from previous grown trees and then each tree is fit on a modified version of the dataset. The algorithm has 3 different tuning parameters: the number of trees (that usually is chosen via cross-validation to avoid overfitting), the shrinkage parameter lambda that determines how fast the algorithm learns (usually it is a small number and the smaller it is the bigger is the number of trees) and then the number of spits each tree must have and, on this factor, depends the complexity of trees. At the end when predictions must be made, each tree gives as an output a value for the response variable and then the average of the values of all the trees is taken as forecasted value. As objective function of our XGB model has been put the reg:sqaurederror function that is the function that aims to minimize the following quantity:

$$\frac{1}{n}\sum_{t=0}^{n}(A_t - F_t)^2$$

that means minimizing mean of the squared difference between the actual value and the forecasted value of each observation.

3.4 Elastic Net

The last method used is elastic net. It is a regularization method that linearly combines the L1 and L2 penalties of lasso and ridge regression. It is able to overcome the criticalities of lasso. Lasso, indeed, performs bad when dealing with datasets where the number of variables is bigger than the number of the rows and, moreover, when dealing with variables that are highly correlated between themselves, it picks only one variable for each group. According to Corporate Finance Institute (2021), "to eliminate the limitations found in lasso, the elastic net includes a quadratic expression (||β||2) in the penalty, which, when used in isolation, becomes ridge regression. The quadratic expression in the penalty elevates the loss function toward being convex. The elastic net draws on the best of both worlds – i.e., lasso and ridge regression". The method undertaken for finding the elastic net regression estimators is divided in two different steps involving both ridge and lasso: in the first stage the ridge coefficients are found and then a lasso shrinkage is applied.

The estimators of elastic net are calculated according to this formula:

$$L_{enet}(\hat{\beta}) = \frac{\sum_{i-1}^{n}(y_i - x_i'\hat{\beta})^2}{2n} + \lambda\left(\frac{1-\alpha}{2}\sum_{j-1}^{m}\hat{\beta}_j^2 + \alpha\sum_{j-1}^{m}|\hat{\beta}_j|\right),$$

This formula tells us that we want to find the Betas, so the coefficients of the variables, that minimizes that formula.

After having trained our elastic net model, the best values of alpha and lambda turned out be, respectively, 0,333 and 0,050075. Constructing the model with these parameters the selected coefficients for the variables are:

Table (VIII)

VAR	COEFF	VAR	COEFF	VAR	COEFF	VAR	COEFF	VAR	COEFF	VAR	COEFF	VAR	COEFF	VAR	COEFF
(Intercept)	4,53E+07	rx_si	0,00E+00	pmi_ser	0,00E+00	trade_balance	0,00E+00	mcon_na	-3,85E+01	pmgue	0,00E+00	m2	0,00E+00	ppi_dur	0,00E+00
ip_sp	3,53E+04	cpexp	1,85E+04	esi_fr	0,00E+00	turnover_for	4,68E+04	vmumint	0,00E+00	ppiman	0,00E+00	m1	0,00E+00	ppi_cons	0,00E+00
ip_fr	2,78E+05	carreg_alfa	6,29E+01	pmi_serexp	-3,37E+04	excon_na	0,00E+00	ppi_dom	0,00E+00	rx_in	0,00E+00	m2_p	0,00E+00	cpi_ipca	0,00E+00
ip_po	1,96E+04	efs_empexp	0,00E+00	busc_exp_selprice	0,00E+00	rx_uk	0,00E+00	jimp_row	0,00E+00	ppi_sp	0,00E+00	ppi_hk	5,72E-02	cpi_nic	0,00E+00
rmdis	7,75E+04	vendite_noalim	1,41E+05	pmi_priceall	0,00E+00	ip_il	0,00E+00	ppi_int	0,00E+00	cpi_en	-1,64E+04	ppitex	0,00E+00	cpi	0,00E+00
rmeuro3	3,23E+05	ip_dk	3,88E+03	clima_imp_tot	0,00E+00	exg_sa	0,00E+00	ppi_cn	-3,74E+04	ppi_au	0,00E+00	ip_ko	-2,08E+04	cpi_core	0,00E+00
r6m	1,05E+05	ip_jp	5,56E+04	btp	1,79E+05	ppiener	-2,06E+04	ppicloth	-1,53E+05	ppi_gr	-2,55E+04	ppiinv	0,00E+00	wpeindss	0,00E+00
ip_gr	1,01E+04	pmi_priceser	8,65E+04	r10y	1,45E+05	ppi_energ	0,00E+00	ppiint	0,00E+00	ppi_noeuro	0,00E+00	n65	0,00E+00	wpeser	0,00E+00
r10y_gy	0,00E+00	carreg_fiat	-1,37E+01	pmi_serbus	0,00E+00	conris_fa	-1,37E+04	ppi	0,00E+00	m3	0,00E+00	pexgextraue	0,00E+00	wpebank	0,00E+00
mib	0,00E+00	ipcostr_wda	0,00E+00	pmi_costr_price	-1,56E+01	exgextraue	0,00E+00	rx_ro	-2,57E+05	loansh_cons5	0,00E+00	ppi_inv	0,00E+00	wpecom	0,00E+00
carreg_mcom	0,00E+00	efs_emp	-6,30E+03	clima_imp_int	0,00E+00	vendite_alim	0,00E+00	ppi_ko	-1,23E+05	ppiche	0,00E+00	cpi_bd	0,00E+00	wpecons	0,00E+00
rx_ch	0,00E+00	pmi_empser	0,00E+00	consconf_fr	-5,63E+03	baddebt	0,00E+00	ppi_gy	0,00E+00	ppi_us	-8,67E+02	gdebt	0,00E+00	ppimot	-1,31E+05
ip_nw	-1,53E+04	carreg_lancia	0,00E+00	pmi_comp	4,82E+04	ip_be	0,00E+00	loans_nfc6	-8,04E+00	ppi_nodom	0,00E+00	loansh_purch5	0,00E+00	arrhotel_sa	-1,13E-01
loansh_purch1_5	6,80E+02	busc_exp_prod	0,00E+00	pmi_outputall	3,36E+04	rx_ru	0,00E+00	vmuminv	0,00E+00	ppi_mx	0,00E+00	vmumcon	0,00E+00	previous_ip_target	1,34E+03
efs_demexp	-1,06E+04	turnover_dom	4,05E+05	comemp	-7,72E+02	ip_au	0,00E+00	ppi_fr	0,00E+00	pexgue	0,00E+00	cpi_em_ipca	0,00E+00		
pmi_inputpriceser	1,70E+03	clima_imp_con	0,00E+00	exint	0,00E+00	pmnfuel_wb	0,00E+00	ppi_po	-4,87E+03	ppi_nw	0,00E+00	ppifood	0,00E+00		
rmlend	3,94E+05	carreg_lcom	1,37E+01	exg	0,00E+00	ppi_be	0,00E+00	pmg	0,00E+00	ppi_euro	0,00E+00	ppicon	0,00E+00		
rx_sz	0,00E+00	carreg	0,00E+00	rx_tr	0,00E+00	ppi_nl	0,00E+00	vmum	0,00E+00	uscpi	0,00E+00	ppi_nodur	0,00E+00		
rmus3	1,39E+05	pmi_comp_fr	2,75E+04	dowjones	2,54E+01	pmgextraue	-4,40E+01	rx_mx	0,00E+00	pexg	0,00E+00	wpeelec	0,00E+00		
carreg_hcom	6,08E-01	busc_exp_ord	0,00E+00	exinv	9,03E+01	wtrade_cpb	0,00E+00	ppi_sw	0,00E+00	pexport	0,00E+00	ppi_dk	0,00E+00		

4. RESULTS

After creating the models and after performing rolling cross-validation, we must have a look to the results and the performances of the two models.

The output of the two rolling cross validation loops of XGB and Elastic Net are summarized in the following tables. The results will be divided in training set results and test set results:

Training Phase

Table (IX)

	time1	time2	time3	predict1_sa	predict2_sa	predict3_sa	predict1_xgb	predict2_xgb	predict3_xgb	predict1_en	predict2_en	predict3_en	obs1	obs2	obs3
1	nov-20	dec-20	jan-21	103,69	104,38	105,310	99,06	101,96	102,44	99,156	99,826	101,021	100,10	100,60	102,30
2	dec-20	jan-21	feb-21	103,69	104,38	105,310	99,48	98,93	102,62	99,826	101,021	102,401	100,60	102,30	102,40
3	jan-21	feb-21	mar-21	103,69	104,38	105,310	101,13	102,35	103,39	100,904	102,215	102,104	102,30	102,40	102,70
4	feb-21	mar-21	apr-21	103,69	104,38	105,310	103,24	103,28	103,64	102,282	102,039	103,582	102,40	102,70	104,50
5	mar-21	apr-21	may-21	103,69	104,38	105,310	103,19	104,44	103,06	102,066	103,691	104,160	102,70	104,50	102,80
6	apr-21	may-21	jun-21	103,69	104,38	105,310	103,20	102,34	102,45	103,691	104,160	105,354	104,50	102,80	104,00
7	may-21	jun-21	jul-21	103,69	104,38	105,310	103,68	103,73	102,92	104,229	105,504	105,499	102,80	104,00	104,90
8	jun-21	jul-21	aug-21	103,69	104,38	105,310	104,07	104,54	103,72	105,354	105,411	105,230	104,00	104,90	104,70
9	jul-21	aug-21	sep-21	103,69	104,38	105,310	104,36	104,27	104,55	105,499	105,345	104,171	104,90	104,70	104,80
10	aug-21	sep-21	oct-21	103,69	104,38	105,310	103,98	104,13	105,21	105,230	103,875	105,271	104,70	104,80	104,30
11	sep-21	oct-21	nov-21	103,69	104,38	105,310	104,38	104,43	104,13	103,875	105,271	106,475	104,80	104,30	106,30

Table (X)

	time1	time2	time3	mape_sa	mape_xgb	mape_en	RMSE_sa	RMSE_xgb	RMSE_en
1	nov-20	dec-20	jan-21	3,43	0,84	0,99	3,47	0,99	1,02
2	dec-20	jan-21	feb-21	2,65	1,54	0,67	2,73	2,05	0,86
3	jan-21	feb-21	mar-21	1,94	0,62	0,71	2,05	0,78	0,88
4	feb-21	mar-21	apr-21	1,22	0,73	0,55	1,31	0,77	0,66
5	mar-21	apr-21	may-21	1,17	0,26	0,90	1,56	0,32	0,98
6	apr-21	may-21	jun-21	1,19	1,06	1,13	1,27	1,20	1,20
7	may-21	jun-21	jul-21	0,54	1,00	1,14	0,61	1,26	1,25
8	jun-21	jul-21	aug-21	0,46	0,45	0,76	0,50	0,61	0,89
9	jul-21	aug-21	sep-21	0,65	0,39	0,60	0,78	0,42	0,62
10	aug-21	sep-21	oct-21	0,78	0,73	0,77	0,86	0,77	0,83
11	sep-21	oct-21	nov-21	0,69	0,86	0,66	0,86	1,28	0,78

Test Phase

Table (XI)

	time1	time2	time3	predict1_sa	predict2_sa	predict3_sa	predict1_xgb	predict2_xgb	predict3_xgb	predict1_en	predict2_en	predict3_en	obs1	obs2	obs3
12	oct-21	nov-21	dic-21	103,69	104,38	105,310	104,41	104,30	105,43	105,298	106,481	104,535	104,30	106,30	105,30

Table (XII)

	time1	time2	time3	mape_sa	mape_xgb	mape_en	RMSE_sa	RMSE_xgb	RMSE_en
12	oct-21	nov-21	dic-21	0,801	0,70	0,62	1,164	1,16	0,73

Let's first take a look to the training phase.

Looking at the tables (IX and XX) it is possible to better understand how rolling cross validation works; we want to forecast the industrial production at three different times t+1, t+2, t+3 and each time the training set rolls down by one: for example look at the first row where dec-20 is time t+2 and jan-21 is time t+3 , they become time t+1 and time t+2 in the following row meaning that the training set has embedded nov-20.

This table is very useful and gives us different information: we can find the different times at which the forecast has been performed, the forecasted values at each time made by all models (SARIMA, XGB and Elastic Net), the actual values of the observations and at the end the error measures performed by each model each time a forecast is made. Another useful thing that comes out from the table is the weakness of the SARIMA model compared to the other two. Indeed, it is possible to notice that the forecast in each period of time (t+1, t+2, t+3) are the same for each row. This results in not satisfying values of MAPE and RMSE that are easier to see especially in the first rows, where they are bigger than 1, while they decrease as much as we go down in the table.

In order to find the best model to forecast the industrial production, we should look at the RMSE and at the MAPE values of the models; but, before doing this, as explained at the beginning of the paper, we should compare the error measure of the two models with the error measures of the seasonal autoregressive model to check if the Elastic Net and the XGB can be considered reliable. All the error measures of the three models are shown below.

Table (XIII)

SARIMA MAPE	XGB MAPE	Elastic Net MAPE	SARIMA RMSE	XGB RMSE	Elastic Net RMSE
1,34	0,77	0,8	1,4	0,95	0,9

Looking at these values of RMSE and MAPE, we can first notice that are all satisfying measures and it shows that even the autoregressive model could be reliable, but we developed the SARIMA model in order to have a benchmark, so let's use this in the purpose it was created for. Both the XGB and the Elastic Net models, according to the RMSE and the MAPE, commit an error that is almost half of the error performed by the SARIMA model. This means that both models can be considered reliable and eligible to be the "chosen" model to perform industrial production forecasting.

So now that we have proved that both models are reliable, we have to choose which one of the two is the best one. Looking only at the error measures that have been showed above it is not easy to decide because the RMSE of Elastic Net is a bit better than the one of XGB but XGB performs better according to MAPE. To choose which model is the best one we should look at the individual error performed by each model each time they forecasted a value of the response variable, and look at which one of the two has the least spread out error values.

To do this, we should pause once again at table (X). Looking at the MAPE values of the two models, we can notice that both error measures have, more or less, the same spread while looking at the RMSE, it is possible to see how the RMSE of the XGB model is more spread than the RMSE of the Elastic Net model. This is even more stressed out by the fact that the standard deviation of the RMSE of XGB net is two times the standard deviation of the RMSE of Elastic Net (0,48 vs. 0,19).

This result can be seen also in the test phase. What was done in this phase was, even if we had the actual values of the response variable, we hid these values to the algorithms and we let them try to forecast it; and the results that have been obtained in the test set are reliable since they do not differ to much than the ones obtained in the training set. What can be noticed is that also in this case both the XGB and the elastic net algorithms performed better than the SARIMA and so could be said as satisfying algorithms. Looking at the test phase, can be confirmed also the fact that the elastic net model performs slightly better than the XGB, since the values of MAPE and RMSE are smaller than the ones of XGB.

So, to wrap up, we have seen that both the XGB and the Elastic Net models are reliable and can be used to perform the forecasting of the industrial production, since they perform better than the seasonal autoregressive model; but Elastic Net is slightly better than XGB because the standard deviation of its errors is smaller than the one of XGB.

4.1 Sensitivity Analysis

In the paragraph "new sub dataset" we have added the variable of the previous values of the dependent variables that represent the value of the industrial production at time t-1. We did this in order to have a predictor to better consider and include seasonality in our model. But does this variable, and so including seasonality, really improve the performances of the model? To check this, we run again the three models (always using rolling cross validation) in the exact same way as before but having a seasonal predictor less. To show the results we will display three tables as before.

Training Phase

Table (XIV)

	time1	time2	time3	predict1_sa	predict2_sa	predict3_sa	predict1_xgb	predict2_xgb	predict3_xgb	predict1_en	predict2_en	predict3_en	obs1	obs2	obs3
1	nov-20	dic-20	gen-21	103,69	104,38	105,31	99,46	101,56	103,72	91,538	93,512	93,792	100,10	100,60	102,30
2	dic-20	gen-21	feb-21	103,69	104,38	105,31	101,87	104,18	104,06	98,706	99,640	103,084	100,60	102,30	102,40
3	gen-21	feb-21	mar-21	103,69	104,38	105,31	102,48	103,53	103,37	100,580	104,214	103,418	102,30	102,40	102,70
4	feb-21	mar-21	apr-21	103,69	104,38	105,31	103,89	102,35	102,78	105,408	104,572	105,787	102,40	102,70	104,50
5	mar-21	apr-21	mag-21	103,69	104,38	105,31	102,45	102,59	102,62	103,176	104,779	106,882	102,70	104,50	102,80
6	apr-21	mag-21	giu-21	103,69	104,38	105,31	102,91	102,85	102,98	104,536	106,657	106,679	104,50	102,80	104,00
7	mag-21	giu-21	lug-21	103,69	104,38	105,31	102,73	102,77	96,17	106,610	106,622	105,478	102,80	104,00	104,90
8	giu-21	lug-21	ago-21	103,69	104,38	105,31	102,62	95,78	103,22	103,720	102,671	102,684	104,00	104,90	104,70
9	lug-21	ago-21	set-21	103,69	104,38	105,31	103,94	104,05	103,97	102,863	102,933	99,445	104,90	104,70	104,80
10	ago-21	set-21	ott-21	103,69	104,38	105,31	105,13	104,16	104,21	104,146	100,663	103,080	104,70	104,80	104,30
11	set-21	ott-21	nov-21	103,69	104,38	105,31	104,73	104,90	108,11	100,923	103,289	103,351	104,80	104,30	106,30

Table (XV)

	time1	time2	time3	mape_ sarima	mape _xgb	mape_ en	RMSE_ sarima	RMSE_ xgb	RMSE_ en
1	nov-20	dic-20	gen-21	3,43	0,99	7,97	3,47	1,06	8,08
2	dic-20	gen-21	feb-21	2,65	1,57	1,72	2,73	1,62	1,93
3	gen-21	feb-21	mar-21	1,94	0,64	1,38	2,05	0,76	1,50
4	feb-21	mar-21	apr-21	1,22	1,15	2,00	1,31	1,33	2,18
5	mar-21	apr-21	mag-21	1,17	0,75	1,57	1,56	1,12	2,38
6	apr-21	mag-21	giu-21	1,19	0,85	2,12	1,27	1,09	2,71
7	mag-21	giu-21	lug-21	0,54	3,19	2,26	0,61	5,09	2,69
8	giu-21	lug-21	ago-21	0,46	3,81	1,44	0,50	5,40	1,74
9	lug-21	ago-21	set-21	0,65	0,78	2,91	0,78	0,82	3,46
10	ago-21	set-21	ott-21	0,78	0,37	1,88	0,86	0,45	2,51
11	set-21	ott-21	nov-21	0,69	0,78	2,48	0,86	1,10	2,87

Table (XVI)

SARIMA Mape	XGB Mape	Elastic Net Mape	SARIMA RMSE	XGB RMSE	Elastic Net RMSE
1,34	1,35	2,5	1,45	1,8	2,9

Test phase

Table (XVII)

	time1	time2	time3	predict1 _sa	predict2_ sa	predict3_ sa	predict1_ xgb	predict2_ xgb	predict3_ xgb	predict1 _en	predict2_ en	predict3 _en	obs1	obs2	obs3
12	ott-21	nov-21	dic-21	103,69	104,38	105,31	104,69	107,47	105,23	106,590	106,847	102,434	104,30	106,30	105,30

Table (XVII)

	time1	time2	time3	mape_ sarima	mape _xgb	mape_ en	RMSE_ sarima	RMSE_ xgb	RMSE_ en
12	ott-21	nov-21	dic-21	0,80	0,51	1,81	1,16	0,71	2,14

Looking at the two tables (XIV and XV) one thing is very clear: we have a big downfall for what concerns the performances of the XGB and Elastic Net model. The errors, indeed, are way bigger than the case in which the seasonal predictor is included in the model. For example, look at the first row where elastic net performs extremely bad committing an error of nearly 8%

every time it makes a prediction, or also lines 7 and 8 where XGB has a RMSE bigger than 5. Other two signs that suggest the unreliability of the two models are the fact that the error measures are quite spread and with high standard deviation, and that they are almost always bigger than one while in the previous analysis they were all smaller than one.

Regarding the SARIMA model can be noticed that the same considerations can be made about the predicted values and the error measures: they are the same as before. This was kind of an expected thing because the SARIMA model, since it is an autoregressive model, it already takes into consideration past values of the response variable in itself, so adding or not the variable y_{t-1} as a predictor doesn't affect the performances of the model. Therefore, can be concluded that the SARIMA model doesn't seem to be impacted by the use of the seasonal predictor.

A different result can be observed regarding the other two models. Their error measures are way higher than the ones committed in the previous case considering the values of the industrial production at time t-1; they are, except for the MAPE of XGB, even higher than the errors committed by the SARIMA and this means that are not reliable. This consideration finds confirmation in the third table of the training phase (table XVI).

As in the case before, the test phase confirms what stated previously. Except for the MAPE of the eXtreme Gradient Boosting, that in this case turned out to be smaller, the case in which the seasonality of the model in considered, and the values of SARIMA that, as said before, are not influenced by the removal of the variable y_{t-1}, all the other values of table XVIII are way bigger than the values of the table XII.

5. CONCLUSIONS

All along the paper we tried to answer the questions made in the second paragraph. We wanted to study the possibility of forecasting the Italian industrial production using a different way than traditional econometric methods. For this reason, we have built two well-known machine learning (ML) algorithms that were ideal to deal with our seasonal data. Beside the implementation of the ML methods has been shown also all work that must be performed before the use of the algorithms and that is fundamental to understand, in the correct way, the dataset and its features and also to eliminate the redundant information, if present, because the more algorithms are simple the better is in terms of performances and overfitting.

From our analysis came out that both our selected models, eXtreme Gradient Boosting and Elastic Net, perform better than the seasonal autoregressive models, making them reliable models that can be used to perform a consistent forecast of the industrial production. But our purpose was the one to choose the best model and looking at the mean average percentual error (MAPE) and at the root mean square error (RMSE) it came out that the elastic net model was the best one, thinking about its performances and the consistency in its errors (measured by the standard deviation). According to these measures, the best model is the Elastic Net; the model that at the same time performs feature selection (indeed some coefficients are 0) and shrinks the coefficient of the variables. But it is not enough to talk about Elastic Net generally, we should point out also the parameters of the model. The paraments that make the Elastic Net perform better than XGB are alpha = 0.333 and lambda = 0.050075. Knowing these parameters is important because we can recreate the model and apply it being sure that we are using the best possible model.

Eventually, after this analysis, what can be stated is that we are now facing a period of time in which technology is everyday improving and evolving supplying us more and more powerful tools. We have just seen a practical examples in which machine learning methods perform better than usually used auto regressive methods like the SARIMA one or ordinary least squares method. Thus far machine learning is the most powerful tool we could have in our hands and maybe, who knows, one day, machine learning tool will be outdated by other technology-based tools.

.

ACKNOWLEDGEMENTS

I would like to acknowledge the following people that, even if in different ways, have been all fundamental in achieving my purpose.

First of all, I would like to thank SOGEI and Prof. Francesco Iafrate for having helped me during the put in writing of my thesis and for having made me keen to R and to the data science.

I am also sincerely grateful from the deep of my heart to my parents, my sister, my grandparents and to my all relatives that have always been on my side and supported me during my years far from home, helping me when I needed and driving me to become a better person.

Finally, I would like to thank Elena, all my friends, inside and outside the university, and my basketball teammates for having made special these last three years.

BIBLIOGRAPHY

- *mice function - RDocumentation*. (n.d.). Mice Function. Retrieved May 31, 2022, from https://www.rdocumentation.org/packages/mice/versions/3.14.0/topics/mice
- *Lasso regression*. (2018, September). https://academic.oup.com/bjs/article/105/10/1348/6122951?login=false
- James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). *An Introduction to Statistical Learning: with Applications in R (Springer Texts in Statistics)* (2nd ed. 2021 ed.). Springer.
- Hyndman, R., & Athanasopoulos, G. (2018). *Forecasting: principles and practice* (2nd ed.). OTexts.
- *Autoregressive Integrated Moving Average (ARIMA)*. (2021, October 12). Investopedia. Retrieved June 15, 2022, from https://www.investopedia.com/terms/a/autoregressive-integrated-moving-average-arima.asp#:%7E:text=ARIMA%20Parameters&text=p%3A%20the%20number%20of%20lag,order%20of%20the%20moving%20average.
- *Forecast Evaluation for Data Scientists: Common Pitfalls and Best Practices*. (2022). https://arxiv.org/pdf/2203.10716.pdf
- *Estimates of energy consumption in China using a self-adaptive multi-verse optimizer-based support vector machine with rolling cross-validation*. (2018). https://reader.elsevier.com/reader/sd/pii/S0360544218305292?token=190FED9A0D5D6D658EDD40902F7E661CD945E9A9E38FCD640D1CF1D970B48500DA0DE85D3DF67E35598F00A1D8316E43&originRegion=eu-west-1&originCreation=20220615130251
- *Comparison of ARIMA model and XGBoost model for prediction of human brucellosis in mainland China: a time-series study*. (2020). https://bmjopen.bmj.com/content/bmjopen/10/12/e039676.full.pdf

- Corporate Finance Institute. (2021, September 1). *Elastic Net*. Retrieved June 16, 2022, from https://corporatefinanceinstitute.com/resources/knowledge/other/elastic-net/
- Thieme, C. (2022, March 30). *Understanding Linear Regression Output in R - Towards Data Science*. Medium. Retrieved August 9, 2022, from https://towardsdatascience.com/understanding-linear-regression-output-in-r-7a9cbda948b3
- Prabhakaran, S. (2022, April 4). *KPSS Test for Stationarity*. Machine Learning Plus. Retrieved August 14, 2022, from https://www.machinelearningplus.com/time-series/kpss-test-for-stationarity/
- *LA DINAMICA DEL DEBITO PUBBLICO IN ITALIA: UN'ANALISI EMPIRICA (1958–2013)*. (2015, July). https://deliverypdf.ssrn.com/delivery.php?ID=821122096002080071125100006030108077026033026084001028104111023006116007028067064097021029123061020113114095116001009001081118046073006036059119004124126086010122102089052024067013118082102126064085064120085110095100028081068120065080024068064118124074&EXT=pdf&INDEX=TRUE
- *Forecasting GDP Growth Using Artificial Neural Networks*. (1999, January). https://www.bankofcanada.ca/wp-content/uploads/2010/05/wp99-3.pdf
- *Business survey data: Do they help in forecasting GDP growth?* (2005). https://pdf.sciencedirectassets.com/271676/1-s2.0-S0169207005X00644/1-s2.0-S0169207004001086/main.pdf?X-Amz-Security-Token=IQoJb3JpZ2luX2VjEGsaCXVzLWVhc3QtMSJHMEUCIHu13j96wOmfHcniVmsqFI4iDbIPCQNesECLI3qay6D9AiEA98h%2FtOglGZua3kIZPG4H2Bv%2BomeTF4UXHIs%2BVcULWeAq2wQI1P%2F%2F%2F%2F%2F%2F%2F%2F%2F%2FARAFGgwwNTkwMDM1NDY4NjUiDNR9ucsb0FjWwb8fZyqvBBex2TCsHyWqpySzUYfct2hdnusaw0mBi6pzQP%2FAA609QzcTSCwbuZT2o3sw02mQGI85Imut9fRC%2BUMIRdYAxF3H9si9E3

k1yrZgajj80SS39dcKIH3l4GtrfCeHljJJLotocuCft9Xqm1wmjBp1mNUYQ0enSaGo2AK5Ip
Us9uGAIubzazH602lNn4ZjTYuOs5Q27M%2B0CTCy0DQqspypk1XCaPINgz0QlkNmV
W6lXKepV%2B9KSt%2BKZhmJTlIiACmUDLvuc%2FirgS3AfHAPYU6lEgXbsVFuMe3
qTBrHvt5veh3orIi4TJkcnDTCwr2W1BpCce1m168ck25GlAOYW%2F8ADzAU3YDx8KN
FaLWbaJeM4yfqjOF%2BEYOCcCQfVLrhZErAi0qO75FcfgQMQTvSgwnaDnoEQVfjdpV
WT2%2Fcp4717I2Q6Xi5oXviLUvTzBP4KXq8oUFHfp%2FUggEwVckCyw%2B5Fwy9qg
DmIR1dUZa5iilJRqfPdinA8w0cF2xKo%2BtuRCMiLbowFm87N7YfLmbKt5K5tgbiq%2B
YJHlT0HZsX60SLOm3Gz%2FmcXAsiRCPNTy6Q8BI2IGGuAt%2Bk%2FACeS7ID25mJ
f13M%2FFRR5wY%2BPlM0azMHizcu2AjQxgk5DPXkIoArgM5rENK9hgb6Yt5QiQ1ciV
%2FN0cd37jzzXyG%2BZzSLMKcGVsAXbnHf3aQ6DWm7cveLQvt9fytlh9hu63hd6iX2K
3UqXLZbIf%2Fg416g6VmVxHdgBKkw6OLtlwY6qQGt9DYdWS55W9OHZeWDOS01be
GkXh62Bc6nmKP7Khfa872D3HjYDZkZwGZkufZjvQNfOaYBzDcb%2Bf9f0xV8h9mRgq
XXHvBqCuzwqTFdC84LUlZVX11FXsmaKZLYCPIPXxQzV2howjj2l14lP0FNs8ZTYl0G
EWthNPX%2FLaFQwFnzr%2F3LZzxEqfykXyeSYAR4XkQSccnHsftStaxIAcnFp7XKOfY
vNr6HOprf&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-
Date=20220816T103002Z&X-Amz-SignedHeaders=host&X-Amz-Expires=300&X-Amz-
Credential=ASIAQ3PHCVTYUORJN2XV%2F20220816%2Fus-east-
1%2Fs3%2Faws4_request&X-Amz-
Signature=8df01b2a69a6df9ee091981b865b9b01bec06e05a276f1eea7fb8b2c5bf95b8b&has
h=2323d5d654796fa51ecc2d99f892d07aab0a467a90478a65715b50e91e09a101&host=6804
2c943591013ac2b2430a89b270f6af2c76d8dfd086a07176afe7c76c2c61&pii=S01692070040
01086&tid=spdf-51632e34-1039-425b-bfb8-
a1d708fdd576&sid=098a6f9f4b552843d35a47d3717bc3638d35gxrqb&type=client&ua=4d
55075655075301555d&rr=73b97e0a687c0e26

- *PREVISIONI AI TEMPI DEL CORONAVIRUS*. (2020, May). http://www.rivistacorteconti.it/export/sites/rivistaweb/RepositoryPdf/2020/novita/2020_06/Bancaitalia_Previsioni.pdf
- *Forecasting industrial prodction in the Euro area*. (2000). https://link.springer.com/content/pdf/10.1007/s001810000032.pdf
- *Forecasting UK Industrial Production with Multivariate Singular Spectrum Analysis*. (2013). https://onlinelibrary.wiley.com/doi/epdf/10.1002/for.2244