Université Catholique de Louvain

LOUVAIN
School of Management

LLSMS2224 - Forecasting
Professor: Bertrand Candelon

Final project

Clarisse Ruchat - 46572000
Axel Vossen - 43682000

Academic year 2023-2024
January 7, 2024

Contents

1 Introduction

This report aims to clarify time series forecasts to provide useful information that will help to make sound decisions for the next three years.

In this project, we take a close look at the real GDP in the Netherlands, using the Box-Jenkins ARIMA method to analyse its trends based on historical data. Our work is not limited to making calculations; we also seek to understand other key variables that could influence GDP, based on reliable data, mostly coming from the St. Louis Federal Reserve. Our goal is not to predict the future perfectly, but to draw a picture of what tomorrow might look like, focusing on our predictions rather than on the technical details of statistical tests.

In our study, we take forecasting a step further by examining how different economic factors, represented by some ten variables, interact and together influence GDP in the Netherlands. Using VAR-X and VECM models, we explore how these various elements may together shape the country's economic future over the next three years.

Our report concludes with a section on innovation, where we put into practice advanced forecasting methods, including Kalman filter and Artificial Neural Network (ANN) models, which represent the state of the art in time series analysis. This section is not just a series of predictions; it merges theory and practical application, detailing the techniques used and comparing the performance of different forecasting models. We take a close look at the influence of the pandemic by separating the periods before and after COVID-19, recognising how this event reshaped economic dynamics.

Combining our observations, we will carry out an exhaustive evaluation of the different forecasts using precise statistical analyses.

2 Methodology

Throughout this work, we will be forecasting the Netherlands' Real Gross Domestic Product for the next three years. We will proceed step-by-step to obtain a complete and clear report.

First of all, we will start by doing a basic analysis of our data and, more specifically, we will test the stationarity of Dutch GDP and remove outliers due to Covid-19 crisis if necessary. As it is not always possible to remove all outliers, we may need to carry out an analysis to see whether or not they have an impact on our results. Next, we will create an ARIMA model to obtain our forecasts.
The most efficient method is to use the Box-Jenkins approach and check whether the residuals are white noise. To do this, we will carry out several statistical tests, such as the Jarque-Bera test for normality or an autocorrelation test, for example. Of course, all the details of our analysis are in the following pages. Next, we can make univariate predictions using the model we have estimated. To illustrate our analysis, we plot a few graphs for each point. To conclude this first part, we will add several potential causal variables and explain their relationship with Dutch RGDP and why we chose them.

Secondly, we will discuss two different models for multivariate forecasting: the VECM model and the VAR model. To this end, we need to identify the additional causal variables and, again, explain their relationship with RGDP. Next, we will need to estimate the optimal lag order to build our model. And as with the univariate forecast, we will perform a few tests before forecasting the estimated VAR model. The Granger-Causality test between variables will be implemented. Then we will determine whether the model residuals are white noise.
In addition, we compare the performance of the prediction model with that of the ARIMA model, and then present an important and useful application of the VAR model. We then estimate the VECM model and finally use the model to plot forecasts of Dutch GDP.

To conclude the analytical part, we will discuss how include innovation forecasts our data. In this section, we will use the Kalman-Filter model to smoothened past variables and describe how it works. Afterwards, we will use the filtered values from Kalman-Filter to train our ANN model that will then return a forecast for the next 3 years.

Finally, we end our report with the limitations or challenges and the conclusion which summarises the main points of the report. We also add our R outputs and a short bibliography with the interesting references we have used.

3 Part I - Univariate Forecast

3.1 Time series analysis

The general trend appears to be upwards until 2020, indicating economic growth over time. There are also significant fluctuations. The first around 2008-2009, which may correspond to the global economic crisis followed by a recovery, and the second, more significant in 2020 due to the impact of the COVID-19 pandemic. After these falls, the series rises again, indicating a potential recovery. This graph does not reveal any obvious seasonal trends. However, more detailed analysis may be required to confirm this, particularly for data that could mask seasonality on this scale. This time series can be viewed on the Federal Reserve Bank of St. Louis website : `https://fred.stlouisfed.org/series/CLVMNACSCAB1GQNL`

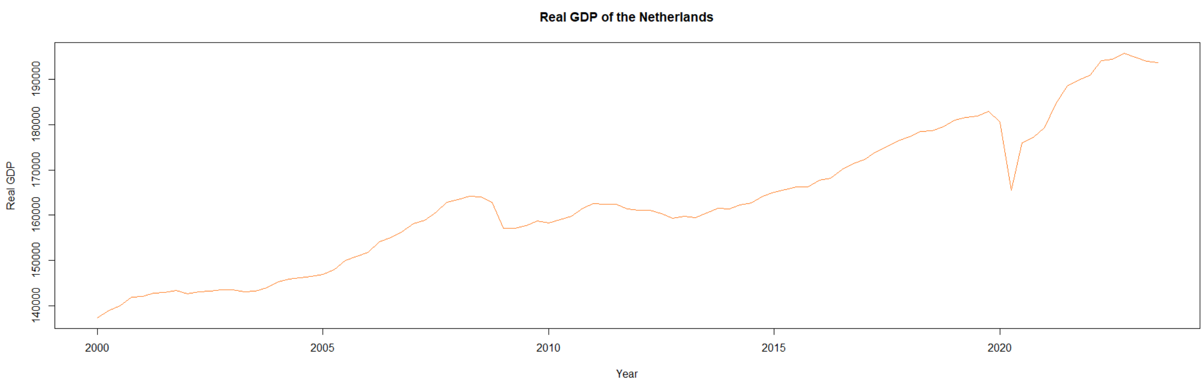


Figure 1: Evolution of the RDGP

3.2 Stationarity test for the ARIMA model

Before starting our ARIMA analysis, we need to be sure that the time series is stationary. From the previous graph, we know that the existence of an upward trend followed by a significant decrease suggests that the series is not stationary. To confirm this, we can perform an Augmented Dickey-Fuller (ADF) test.

Tested hypothesis:

$$H_0 : \textit{the data are non-stationary} \tag{3.1}$$
$$H_1 : \textit{the data are stationary} \tag{3.2}$$

We obtain the following results:

```
  Augmented Dickey-Fuller Test

data:  gdp2k_ts
Dickey-Fuller = -2.6997, Lag order = 4, p-value = 0.2875
alternative hypothesis: stationary
```

The p-value of the test is greater than 0.05, which suggests that we cannot reject the null hypothesis at the 5% significance level. In other words, there is insufficient statistical evidence to conclude that the time series is stationary. In practice, this means that Dutch real GDP could have a trend or random walk pattern that would require differentiation to make the series stationary before proceeding with ARIMA analysis or other time series forecasting methods.

To obtain stationary data, we will try converting our time series into its difference of logarithm and thus explore the growth of the RGDP. Afterwards, we have compute the ADF test again to check if the time series became stationary. We therefore take the logarithm of the differences of the RDGP and repeat the test. We now obtain :

```
  Augmented Dickey-Fuller Test

data:  gdp2k_ld
Dickey-Fuller = -4.7223, Lag order = 4, p-value = 0.01
alternative hypothesis: stationary

Warning message:
In adf.test(gdp2k_ld) : p-value smaller than printed p-value
```

The p-value is less than 0.05, so we can reject H_0, which suggests that the series is stationary and has no unit root, which is a prerequisite for ARIMA modelling.

To be sure of our hypotheses, we carried out a final test called the KPSS Unit Root Test in order to verify the results from the Augmented Dickey-Fuller test. We define the KPSS unit root test with stationarity as null hypotheses.

```
#######################
# KPSS Unit Root Test #
#######################

Test is of type: mu with 3 lags.

Value of test-statistic is: 0.0496

Critical value for a significance level of:
                10pct  5pct 2.5pct  1pct
critical values 0.347 0.463  0.574 0.739
```

The critical value 0.0496 is lower than all the critical values listed (0.347, 0.463, 0.574, 0.739), which means that the time series is stationary at all common significance levels (10%, 5%, 2.5%, 1%). Furthermore, there is no evidence of a unit root, which implies that the time series is stationary.

Finally, we performed a Jarque-Bera test to determine whether or not the series is normally distributed. Since we obtained a very low p-value (2.2e-16), this suggests that the null hypothesis of normality is rejected and that the data do not follow a normal distribution.

We can plot our new stationary data:

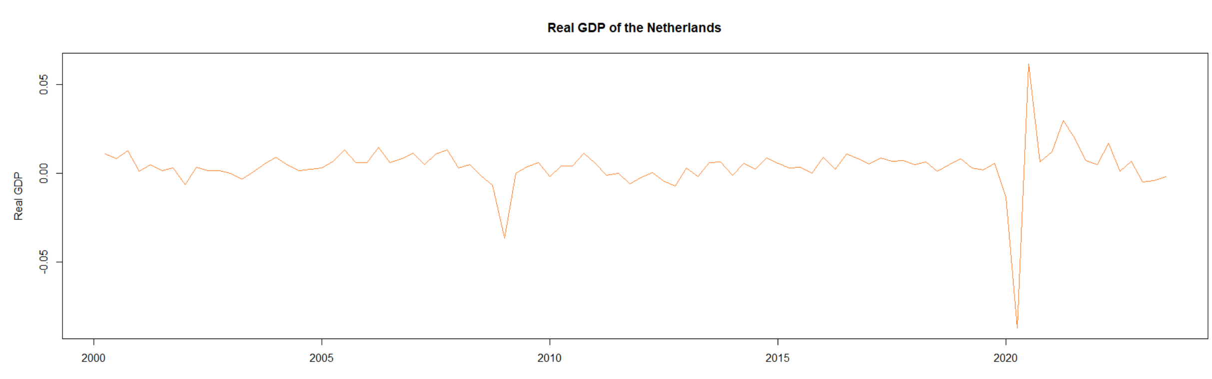


Figure 2: Stationary evolution of the RDGP

We are now seeing constant fluctuations around 0, indicating that the growth rate is changing but with no clear trend or seasonality. On the other hand, we can also observe that extreme values in 2008 and 2020 for the RGDP are represented as well. We will thus move to the next section to explore how to deal with those extreme values.

3.3 Detection of outliers

By using the function **'boxplot'**, we can see if our data have outliers or not. If they do, they will be visible as small dots beyond the confines of the box:

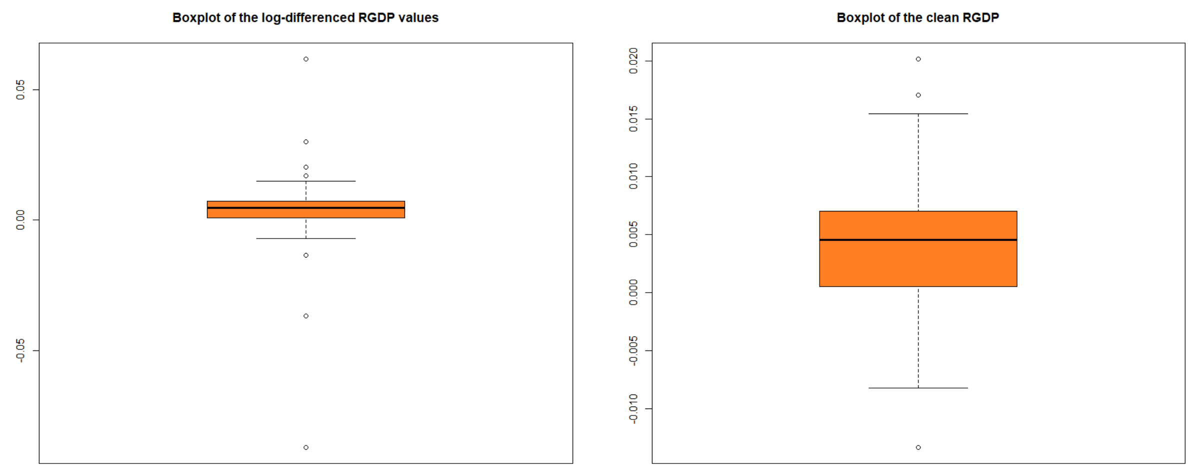


Figure 3: Boxplot function for the RGDP and the clean RGDP

We observe that the log difference of the real gross domestic product for the Netherlands (the left graph) has a median close to zero and contains 7 outliers, which appear as small dots in the box plot. These outliers are individual values that fall outside the usual range and may represent atypical variations in the data. It is important to eliminate these outliers as much as possible, as they can influence the results of statistical analyses.

The new box plot after applying the **'tsclean()'** function (the right graph) shows that the data has been cleaned to some extent, as there are fewer outliers. The central box is narrow, which shows that the values for 50% of the data are quite similar to each other. However, the presence of outliers above and below the box indicates that there are still extreme values in the data. The median is indicated by the line inside the box and appears close to zero, suggesting that the central tendency of the cleaned data is near-zero variation in the log-differentiated GDPR values.
As we cannot remove all outliers, we compared the mean, median and variance with and without them to know if they had a significant impact on the results.

```
With outlier: Mean = 0.003650974  Median = 0.004802973
Without outliers: Mean = 0.004037562  Median = 0.00475647
```

Based on this results, it seems that outliers have not significant impact on the mean and median of the series. This suggests that the central tendency of the data is not drastically altered by the presence of outliers.

3.4 Building the model

Once we have tested everything to see which model we should use, we can use the ARIMA model for our data. This will help us to automatically identify the p,d,q parameters of the model.

The ARIMA model is best suited for several reasons. We have shown that the Dutch real GDP time series is a non-stationary model with an upward trend followed by a significant decline, suggesting the need for differentiation. In addition, the first Augmented Dickey-Fuller (ADF) test confirmed non-stationarity with a high p-value. After differentiation and registration of the series, the ADF test gave a lower p-value, indicating

stationarity, suitable for ARIMA modelling. The KPSS test also confirmed this hypothesis, showing that the series was stationary at common significance levels. Although the Jarque-Bera test indicates non-normality, the ARIMA model is robust to such violations because it focuses on the autocorrelation structure of the data rather than its distributional assumptions.
To develop an appropriate ARIMA model, we will employ the Box-Jenkins method, starting by determining the necessary AR and MA orders through analysis of ACF and PACF plots.

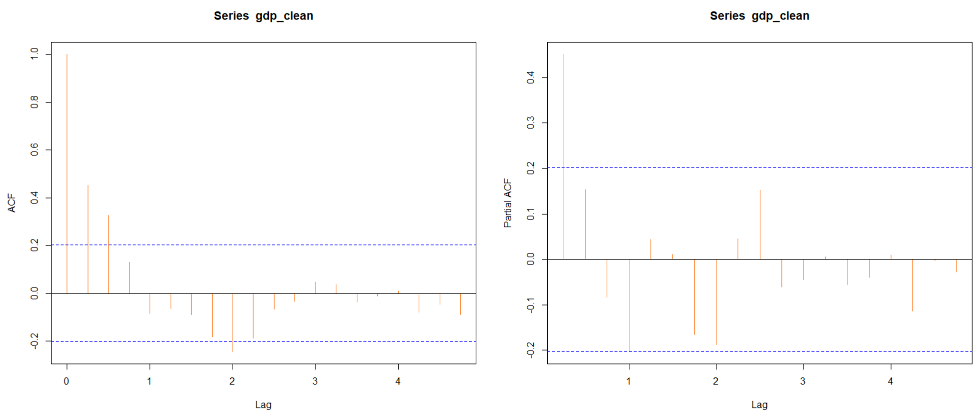


Figure 4: Autocorrelation and Partial Autocorrelation functions

Looking at the ACF and PACF plots, we see that the first significant value is at lag 0 of the autocorrelation function, while the last significant value of the partial autocorrelation is at lag 1.This means that the additional past values have no influence on the real value of the series once the effect of the first lag has been taken into account. Given the data, an AR(1) model might be appropriate to model this series for our first analysis, so we would have an ARIMA(p,d,q) with p being 1. Although, we must also keep an eye on the seond lag since the PACF is almost significant for this lag. We then need to validate this first approximation of the model using R's **'auto.arima'** function.

```
Series: gdp_clean
ARIMA(2,0,0)(0,0,2)[4] with non-zero mean

Coefficients:
         ar1     ar2     sma1     sma2   mean
      0.3527  0.2618  -0.3287  -0.2045  4e-03
s.e.  0.1012  0.1030   0.1144   0.1125  6e-04

sigma^2 = 2.379e-05:  log likelihood = 368.94
AIC=-725.88   AICc=-724.91   BIC=-710.62
```

Finally, we have an AR(2) instead of AR(1) as we had previously predicted. Apparently our time series still is impacted by seasonality, that can be explained by various trends in the gdp fluctuation i.e. with the holidays We can now write the model as:

$$y_t = 0.00142 + 0.3527 y_{t-1} + 0.2618 y_{t-2} + \varepsilon_t \tag{3.3}$$

Where y_t is the current value of the times series RGDP, y_{t-1} and y_{t-2} are the first and second lags of the time series respectively and the constant is equal to $0.0004 \times (1 - 0.3527 - 0.2618)$.

3.5 Testing the model

It is important to test the model with misspecification tests to ensure that the one used is the right one.

3.5.1 Normality

To test if a model is normally distributed or not, we can use the Jarque-Bera test:

$$H_0 : \text{S} = 0, \text{K} = 3$$
$$H_1 : \text{S} \neq 0, \text{K} \neq 3$$

```
  Jarque Bera Test

data:  ARIMA_gdpcl$residuals
X-squared = 2.9144, df = 2, p-value = 0.2329
```

The test shows that the p-value is 0.2329, which is higher than the common significance level of 0.05. We cannot reject the null hypothesis and can conclude that the residuals are normally distributed.

3.5.2 Homoscedasticity

Secondly, the Breusch-Pagan test is useful for detecting heteroscedasticity in a model:

$$H_0 = \text{Data are homoscedastic}$$
$$H_1 = \text{Data are heteroscedastic}$$

```
  studentized Breusch-Pagan test

data:  ARIMA_gdpcl$residuals ~ fitted(ARIMA_gdpcl)
BP = 0.26919, df = 1, p-value = 0.6039
```

The p-value is 0.6039, which is well above 0.05, and therefore does not allow the null hypothesis to be rejected. This means that the residuals in our ARIMA model are homoscedastic, and therefore that there is no apparent change in variance in the residuals.

3.5.3 Autocorrelation

Finally, to test our model we can use the Breusch-Godfrey test to know if our model have autocorrelation. The hypotheses for the following test are:

$$H_0 = \text{No autocorrelation}$$
$$H_1 = \text{Evidence of autocorrelation}$$

```
  Breusch-Godfrey test for serial correlation of order up to 1

data:  ARIMA_gdpcl$residuals ~ 1
LM test = 0.00025602, df = 1, p-value = 0.9872
```

Once again, as the p-value is 0.9872, there is no evidence to suggest the presence of autocorrelation in the residuals, we can say that is is white noise. The residuals seem independent of each other.

We can confirm our hypotheses with a Box-Ljung test, which gives the same results:

```
  Box-Ljung test

data:  ARIMA_gdpcl$residuals
X-squared = 0.13278, df = 4.5433, p-value = 0.9992
```

Conclusion

According to the tests carried out, the residuals of the ARIMA model satisfy the various hypotheses, they are normally distributed and homoscedastic, which indicates that there is no apparent variation in variance over time. Furthermore, there is no evidence of autocorrelation among the residuals, suggesting that the residuals are independent of each other. These results therefore show that the ARIMA model is well suited to the Netherlands RDGP. We can also confirm these hypotheses with the help of *[Appendix 3.5]*, which gives a more global overview.

3.6 Forecasting the model

To forecast the Netherlands' Real Gross Domestic Product over the next 3 years, we can now use our ARIMA(2,0,0). The resulting graph shows the historical trend and future forecasts. The historical data from 2010 shows fluctuations, with a significant drop around 2020, probably due to the impact of the Covid-19 pandemic. The forecast beyond 2020 predicts potential future trends, with a shaded area representing the confidence intervals, i.e. the range within which future values are likely to fall; the darker the area, the higher the probability of having values within this interval.

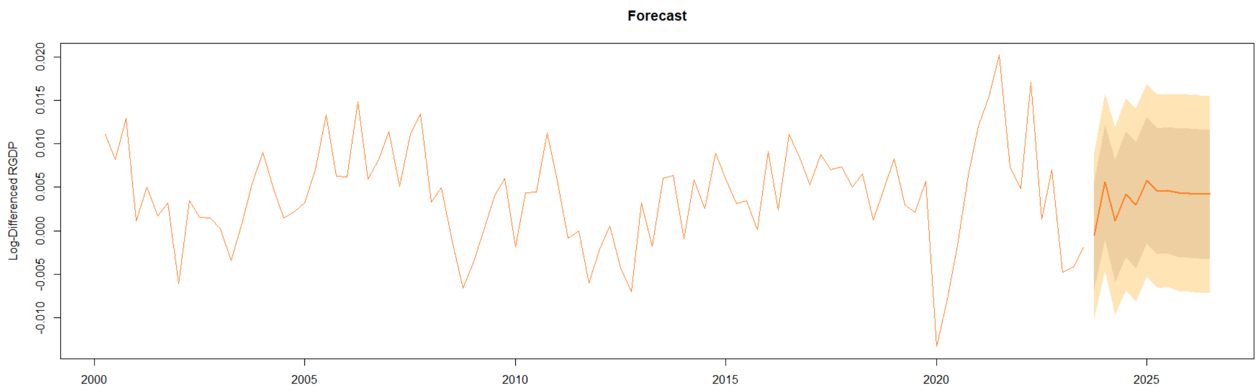


Figure 5: Forecast

Moreover, the forecast table *[Appendix 3.6]* shows predictions for upcoming 12 quarters.
The data provided by the "Point Forecast" column reveal a trend and relative stability in Dutch RGDP. In the fourth quarter of 2023, a slight fall is forecast, while the following quarters, in 2024 and 2025, show slight gradual increases. This suggests a modest and steady recovery in the Dutch economy. The most significant increase is forecasted for the first quarter of 2025, with a slight downward trend thereafter. However, the forecasts are fairly close to zero, which means that there is no significant change expected in GDP growth in the near future. Thanks to the forecast intervals (Lo80, Hi80, Lo95 and Hi95), we can say that the variation remains within a narrow range.

3.6.1 Forecast quality

To check the quality of our forecast, we have used the **'accuracy()'** function with respect to forecasted values. With this function, we can understand that it returns several values corresponding to different accuracy indicators. In general, the closer the value is to zero, the more accurate the model. We can consider that our model is relatively accurate with a Mean Absolute Error (MAE) of 0.0036.

```
                        ME        RMSE         MAE       MPE     MAPE
Training set -2.036996e-05 0.004745512 0.003646687 -249.5885 550.8459
                  MASE          ACF1
Training set 0.5790534 -0.001650334
```

Usually we ignore the MPE and MAPE values since they are not issued from a finite interval. Therefore, when having a value equal or close to zero, it tends to return no value or an infinite one.

3.7 External indicators

The aim of this section is to find some possible causal variables for real gross domestic product in the Netherlands. We did some research to find out which macroeconomic indicators can influence the RGDP. We then searched for the following indicators on the Federal Reserve Bank of St.Louis website before starting our analysis. We were careful to use only data valid until 2023:

- **Producer Price Index (PPI) for Food:** The PPI for food reflects production costs for producers. An increase in the PPI can reduce the profit margin of companies or be passed on to consumers, thus affecting consumption and GDPR. Conversely, a low PPI can stimulate consumption by making food more affordable, thereby supporting economic growth in the Netherlands.

- **Education and Health:** These are two key indicators of economic development. Better health will increase labour productivity, having a positive impact on GDP. A well-educated population encourages innovation and efficiency. These factors strengthen resilience and the potential for economic growth.
- **Retail:** The performance of the retail sector can have a significant influence on GDP in the Netherlands. An increase in retail sales indicates strong household consumption, which stimulates economic growth. Conversely, a slowdown in this sector can mean lower consumer confidence and weaker purchasing power, which will have a negative impact on RGDP.
- **Exports:** This indicator plays a crucial role in the Dutch economy. An increase in exports leads to an influx of income into the economy, stimulating production and employment. It can boost GDP, reflecting a strong economy and competitiveness on the global market.
- **Production:** An increase in output signals a growing economy, where industries are operating at high levels to meet consumer and business demand. This generally translates into higher employment and incomes, stimulating consumption and investment.
- **Imports:** Imports into the Netherlands influence GDP by affecting domestic consumption and the balance of trade. An increase in imports can indicate strong demand from consumers and businesses, thus contributing to economic growth.
- **Unemployment:** A high unemployment rate suggests an economic slowdown, with companies cutting back on production and labour. It can lead to a fall in consumption and investment as household incomes fall and economic uncertainty increases.

3.8 Comparison with ARIMAX

To further explore the impact of such varaibles, we have computed the ARIMAX model for some variables *[Appendix 3.8]*. We decided to make an ARIMAX for five potential causal variables: Exports, Production, Education, Unemployment and Imports. We chose these variables because they are potential causal variables and we are likely to use them later in the VAR model.

Before starting, we carried out the three basic tests for each of them to ensure that they are stationary, that they are normally distributed and that they have no autocorrelation. We found that Production, Unemployment and Imports meet these criteria well. However, exports are not normally distributed while education is not stationary.

Thanks to *[Appendix 3.4]*, we can compare the ARIMA of **'gdpclean'** to the ARIMAX of every additional variables:

- **AIC:** As we have already explained, the AIC is a measure of the relative quality of the model. A lower AIC indicates a better model. In this situation, the ARIMAX model has lower AICs. This is because adding variables will make the model more reliable.
- **Error Metrics:** These are ME, RMSE, MAE, MPE, MAPE and MASE. The lower they are, the more efficient the model will be. In this case, the error measures of the ARIMA model are quite similar to those of the ARIMAX model.
- **Autocorrelation of Residuals (ACF1):** When the ACF1 is close to 0, it indicates the absence of autocorrelation. We have already tested this criterion recently and we can see that all the ARIMAX models for the additional variables have an ACF1 very close to 0, but that the ARIMA model has an even lower ACF1.

In conclusion, if we compare the ARIMA model with the ARIMAX models, it seems that the ARIMA model performs well and even slightly better than the ARIMAX models. However, using an ARIMAX wouldn't have been a bad idea, given that they're still pretty good performers.

4 Part II - Multivariate Forecast

In this section, we will perform a multivariate forecasting analysis by estimating VAR-X and VECM models for the Netherlands over a three-year period.

4.1 Exogenous variables

First, we searched for exogenous variables that could explain the Dutch RGDP to include in our models. There are many indicators available on the FRED site and we must therefore make a choice to keep only the most relevant of them. Therefore, we kept the endogenous variables listed in section 3.7 to which we added the time series listed hereunder.

- **Energy Prices Worldwide (EP):** Energy prices can influence a country's GDP in several ways. Firstly, high energy prices can increase production costs, reduce competitiveness and slow economic growth. Secondly, they can reduce consumers' purchasing power, leading to a fall in consumption. In addition, this variation in prices can affect inflation, which has an impact on monetary policy. Finally, it influences the energy sector itself, which is an important component of the Dutch economy. The transition to renewable energy sources may influence the country's economic structure and competitiveness in the long term.

- **Economic Policy Uncertainty (EPU):** This variable measures the ambiguity surrounding economic policies, which can affect Dutch GDP through their impact on investment and consumption. It remains exogenous, however, as it is often influenced by global factors or policy decisions outside the Dutch economy.

- **Raw Material Price (RM):** The price of raw materials influences the cost of production in the Netherlands and therefore GDP. However, these prices are determined on the world market and are beyond the direct control of the economy, so they remain exogenous.

- **Real Effective exchange rate (REER) for the Euro Area:** The REER is a measure of the relative strength of a nation's currency (Euro) in comparison with those of the nations it trades with. This indicator is valuable to determine whether a nation's currency is undervalued or overvalued.

- **Personal Consumption Expenditure of Energy:** Personal expenditure on energy consumption reflects the energy consumption of Dutch households and directly influences GDP through domestic demand. However, this expenditure is largely conditioned by world energy prices, which makes it exogenous to the Dutch economy.

- **Spot Exchange rate:** This time series gathers the amount one currency will trade for another one at a specific point of time. To fit our domain of research, we have selected the US dollar to Euro spot exchange rate.

- **Oil prices:** Nowadays, a few countries have a near monopoly concerning the oil production. This energy source is still used worldwide for diverse reasons. The Netherlands is not a producing country but must be in some way influenced by the oil price since this energy source must be purchased by various economic players inside the country.

To begin the multivariate forecasting, we performed a series of linear regression analyses to explore the relationship between log-differenced GDPR and various other log-differenced economic indicators.
For each indicator, a separate model is estimated using the function **'summary()'** to display the statistical summary of each regression, which includes coefficients, significance levels and other figures to help understand which indicators may have a significant impact on RGDP.

Afterwards, we computed a multiple linear regression with all the variables listed in the 3.7 and 4.1 subsections. Based on our computations and the output that resulted from our multiple linear regression, we were able to select relevant indicators for our VAR-X model based on the significance of the different estimators with regard to a confidence interval of 95 percent, see *[Appendix 4.1]*. Besides, we also added exogenous indicators to the

model if they positively impacted the adjusted R^2. As shown in *[Appendix 4.1.2]*, The indicators selected to explain Dutch RGDP are as follows: PPI, exports, retail, unemployment, production growth, raw material prices, the REER, SER and PCE. Alltogether, they explain 64.77 percent of our model.
Then, we can predict the variation of the Dutch Real Gross Domestic Product with this following equation:

$$RGDP = 0.0014435 - 0.1697 \times PPI + 0.1406 \times Exports - 0.0401 \times Unemployment + 0.0011 \times Production + 0.1782 \times Retail - 0.0481 \times Rawmaterial - 0.0527 \times REER - 0.0658 \times SER + 0.0587 \times PCE \tag{4.1}$$

4.2 Buidling the Vector Autoregressive Model

So far, we only have considered a unilateral relationship between the different variables and through the different models with the forecast variables being influenced by the predicted variables. In this part, we are going to try and reverse this process with all the variables that affect each other. This is where the vector autoregressive framework (VAR(p)) comes to hand. A VAR model is a generalisation of the univariate autoregressive model for forecasting a vector of time series. It comprises one equation per variable in the system. The right-hand side of each equation includes a constant and lags of all the variables in the system. Note that the dimension of a VAR process is equal to 2 or higher and it is considered as stationary.

4.2.1 Stationary and outliers

Before diving deeper into our VAR model, we must check that the significant indicators found in the previous subsection are indeed stationary and normally distributed. If they do not respect both constraints, they could influence our model in the wrong way. In such case, it could falsen our analysis of the outputs and then of the model. To this end, we used the **adf.test()** and the **ur.df()** functions in order to test the stationarity of our variables by checking for unit roots in our model. Then, we tested their normality with the **Jarque Bera test**.
In addition, to avoid any outliers, we use the **tsclean()** function when the data were not normally distributed. If we obtained a p-value > 0.05, we concluded that the time series was normally distributed. This allows us to test all the additional variables so that they have no wrong impact on the model. Following our tests, we had to exclude PPI and exports indicators due to their non-normal distribution.

4.2.2 VAR Model

The second part of the autoregressive model consists of finding p, which is the number of delays included in the system. The result can be found below, which returns the "p" optimal delays as a function of the information indicator. In this case, the values of p are different. We will consider the value returned for the Akaike information criteria (AIC) and Bayesian Information Criteria (BIC or SC). We usually consider the BIC value rather than the AIC value, as it tends to select a large number of delays. This step is quite important to find the optimal p for the VAR function. If p is too short, the model will be misspecified, while if too large, too many degrees of freedom will be lost. Ultimately, the number of lags must be sufficient for the residuals to be individual white noise.

Note that the VAR model assumes that all the variables used for the model are endogenous. Since the purpose of our analysis is to depict the influence of **exogenous** indicators of the real GDP of the Netherlands, we must account for those indicators as well. Although we used the same function, we added the "exogen =" parameter in the **VARselect()** function with the matrix composed of exogenous variables so that the VAR model accounts for those indicators as well. Below is the output from that determines the value of p:

```
AIC(n)  HQ(n)  SC(n) FPE(n)
    10      1      1      1
```

This output above indicates that when selecting the optimal lag for the VAR model, the AIC suggests using a lag order of 10. This means that incorporating the last ten periods of data will provide the most effective balance between model goodness of fit and complexity, minimising potential over-fitting while capturing the necessary dynamics in the data. But as stated before, the *Akaike information criteria* tends to select large numbers of lags. Therefore, we have taken the SC value for p, which is equal to 1.

4.2.3 Granger causality

Once we have figured the value op p, we must test the causality of our model through the *Granger test*. It states that if a random variable X is granger causal of the random variable Y, the past variables of X contain information to predict Y. When computing the Granger test on R, we assume the following hypothesis:

$$H_0 : \textit{Time series X does not Granger-cause the others} \tag{4.2}$$
$$H_1 : \textit{Time series X Granger-causes the others} \tag{4.3}$$

As we can observe on the *Appendix 4.2.3*, only the Unemployment indicator Granger-causes other indicators with its p-value inferior to the 0.05 threshold. After further computation, we observe that unemployment is only causal for the RGDP.

4.2.4 Testing the VAR Model

Finally, after all those computation with regards to the VAR(1) model, we implemented the portmanteau tests. As a reminder, the Portmanteau tests are statistical hypothesis in which the null hypothesis is well specified, while the alternative hypothesis is more loosely specified. We started with the **serial.test()** function computing multivariate Portmanteau- and Breusch-Godfrey test for serially correlated errors with:

$$H_0 : \textit{no auto-correlation hypothesis} \tag{4.4}$$
$$H_1 : \textit{auto-correlation hypothesis} \tag{4.5}$$

```
	Portmanteau Test (asymptotic)

data:  Residuals of VAR object Varm
Chi-squared = 248.58, df = 240, p-value = 0.3382
```

With a p-value above the 0.05, we cannot reject the null hypothesis. We thus assume no auto-correlation of the residuals of our model.

Secondly, we had to test for homoscedasticity for which we used the **arch.test()** function. As for the Goldfeld and Quandt test, we have:

$$H_0 : \textit{Homoscedasticity} \tag{4.6}$$
$$H_1 : \textit{Heteroscedasticity} \tag{4.7}$$

Here, the p-value is equal to 0.1585. Therefore, we cannot reject the null hyothesis of homoscedasticity.

For the last Portmanteau test, we shall find out if the residuals are indeed Normally distributed in order to fulfill the White noise criteria. To this end, we computed the **normality.test()** function with:

$$H_0 : \textit{Residuals are normally distributed} \tag{4.8}$$
$$H_1 : \textit{Not normally distributed} \tag{4.9}$$

```
  JB-Test (multivariate)

data:  Residuals of VAR object Varm
Chi-squared = 8.7725, df = 8, p-value = 0.3618
```

Since the p-value is above 0.05, we cannot reject the null hypothesis and assume that the residuals are normally distributed, concluding that the residuals of the VAR(1) model are White noise.

4.2.5 Forecast of the VAR(1) model

As a result, we forecasted our model for the next three years (equivalent to 12 quarters). Clearly, the forecast for the Dutch's real growth domestic product will follow a flat progression in the following years.

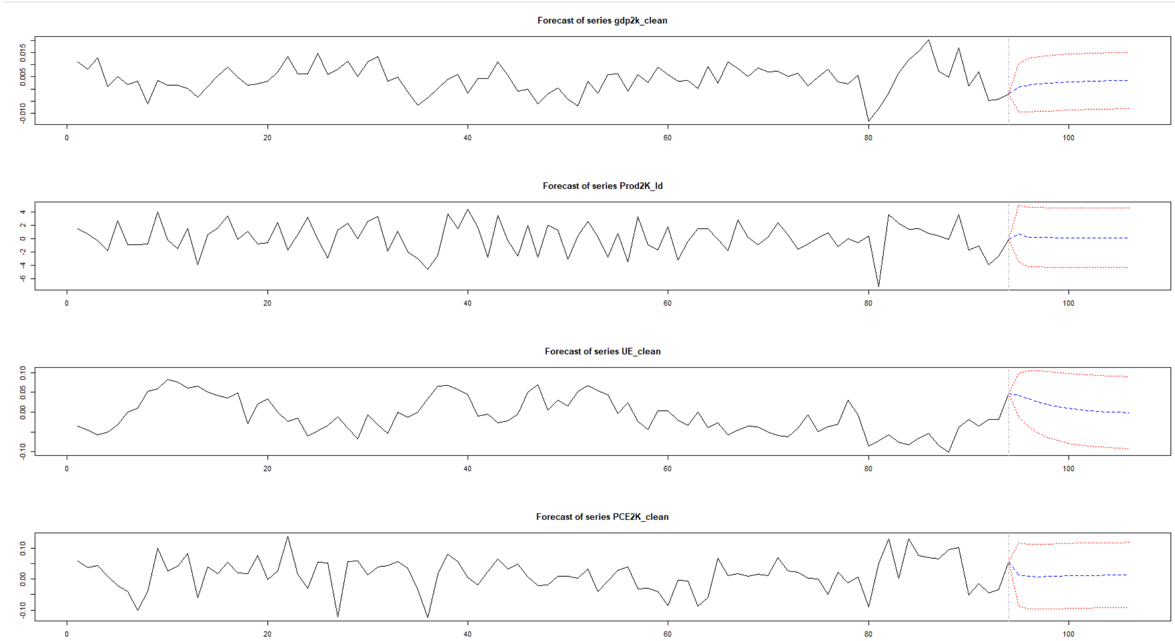


Figure 6: 3y forecast of the VAR(1) model

4.3 VECM model

In this section, we will evaluate the cointegration relationship of our indicators. In a case where indicators are positively cointegrated, they will tend to move in the same direction. Meanwhile, if they are badly, they could evolve in opposite direction and thus deteriorate the model. Following this assumption, a positively cointegrated relationship is valuable to predict the path of an indicator. Based on our previous computations, here are the two hypothesis following the *Johansen procedure*:

$$H_0 : \textit{There are r cointegration factor(s)} \tag{4.10}$$

$$H_1 : \textit{There are r+1 cointegrating factors} \tag{4.11}$$

with $r = 0$ meaning there is no cointegration relationship.

```
######################
# Johansen-Procedure #
######################

Test type: maximal eigenvalue statistic (lambda max) , with linear trend

Eigenvalues (lambda):
[1] 0.50437374 0.33435157 0.27134793 0.07353817

Values of teststatistic and critical values of test:

          test 10pct  5pct  1pct
r <= 3 |  7.03  6.50  8.18 11.65
r <= 2 | 29.12 12.91 14.90 19.19
r <= 1 | 37.44 18.90 21.07 25.75
r = 0  | 64.58 24.78 27.14 32.14

Eigenvectors, normalised to first column:
(These are the cointegration relations)
```

From this result, we can conclude that we are in presence of an r=3 model since the value of the test is higher than the critical value for r=0, r=1 and r=2 in a 5 percent bound. On the other hand, r=3 has its test value under the 5% critical value so we cannot reject the null hypothesis. The reason to this assumption is that for each r, when the test's value is higher than the critical ones, we reject the null hypothesis and move up to the next r value.

Additionally, we computed the **"vec2var()"** function to transform the VCM in differences to a VAR in levels with *[Appendix 4.3]* as the output of the estimation. Finally, we can observe below the 4 time series that were plotted to represent the forecasted values for the next 10 quarters of our VECM model. In this case, we can observe that the forecasts fluctuate compared to the VAR forecast, probably because it accounts for the cointegration relationships.

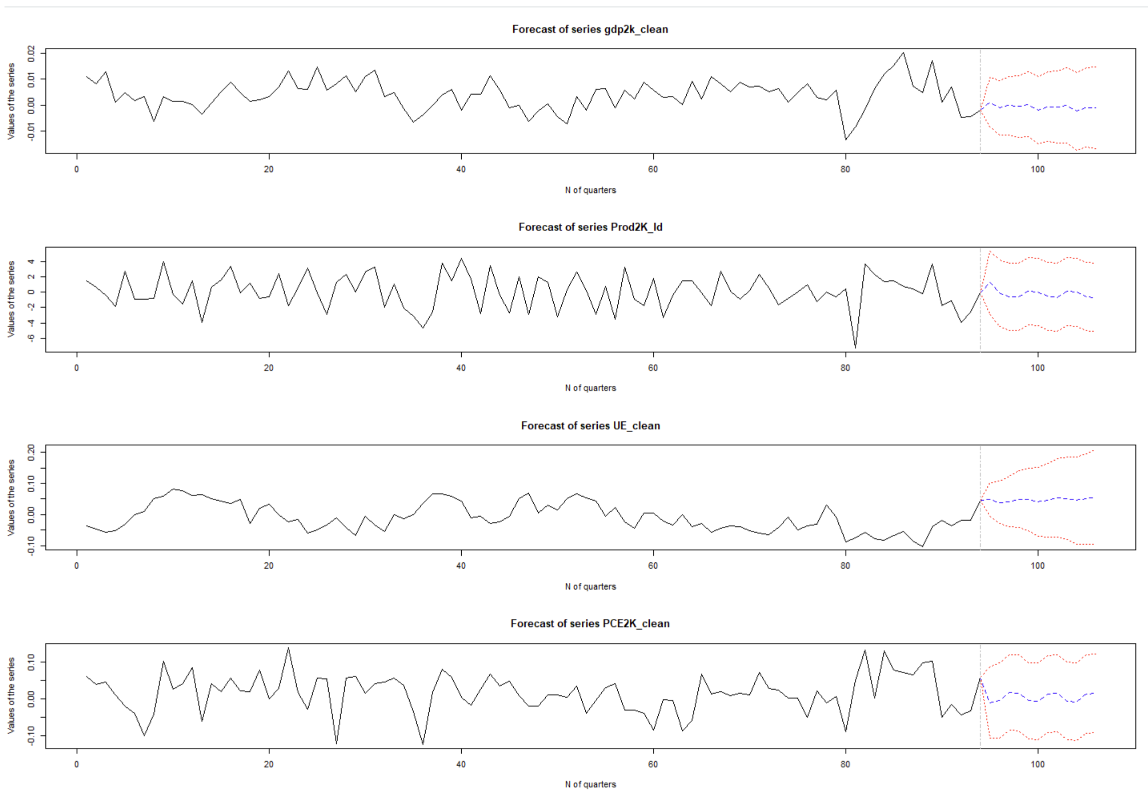


Figure 7: 3y forecast of the VECM model

4.3.1 Test of the VECM residuals

Following the same logic of the VAR section, we have computed the Portmanteau tests in order to check if the residuals are white noise. Starting with the auto-correlation, the output of the **serial.test()** function is:

```
    Portmanteau Test (asymptotic)

data:  Residuals of VAR object vec2var_vecm.test
Chi-squared = 236.84, df = 228, p-value = 0.3301
```

We can conclude thanks to the p-value that the residuals are not auto-correlated.

Then we continue our analysis with the **ARCH** test for heteroscedasticity. Keeping the same hypothesis depicted in the 4.6 and 4.7 formulas, we obtain:

```
  ARCH (multivariate)

data:  Residuals of VAR object vec2var_vecm.test
Chi-squared = 490.66, df = 500, p-value = 0.6087
```

We thus cannot reject the null hypothesis since the p-value equivalent to 0.6087 exceeds the 0.05 threshold. The last test computed is to determine if residuals are normally distributed. Again, we keep the same hypothesis as in the 4.2.4 section. As a result we have:

```
	JB-Test (multivariate)

data:  Residuals of VAR object vec2var_vecm.test
Chi-squared = 5.022, df = 8, p-value = 0.7552
```

Based on the output, we can also assume that the residuals are normally distributed. As a conclusion, the residuals are white noise and it gives us confidence in our model.

4.3.2 Accuracy of the VECM

Finally, we decided to check the accuracy of the VECM forecast in two different ways. First by estimating the forecast of the last 12 quarters present in our database. Then we plotted the estimated values to the actual ones present in the time series for each indicators, see *[Appendix 4.3.1]*.
Besides, we computed a loop of verification based on the residuals of our VECM model and the fitted values in order to use the **accuracy()** fucntion.

```
                        ME        RMSE         MAE       MPE     MAPE      MASE        ACF1
Training set -2.078626e-18 0.004909812 0.003943624 -396.9742 785.8763 0.8051822 0.03359186
```

Since most of the values average 0, we have more confidence in our forecasted VECM model. We can observe that the MPE and MAPE take huge values, but as explained earlier, we will ignore those two values since they lack a finite interval. Moreover, the predicted values ouf our accuracy test in *Appendix 4.3.1.* are very close to the actual values.

5 Part III - Innovation

In an ever-changing economic context, machine learning (ML) approaches are becoming increasingly popular and crucial for decision-makers. In this section, we will integrate a few innovation methods to take what we have already analysed, a step further.

We chose to build a Kalman-Filter model first, but the aim is to finally build an Artificial Neural Network (ANN) model, as it is capable of withstanding changes, which is important in economics. They are also capable of handling situations in which some data is missing and of dealing with time series.

5.1 Kalman-Filter Model

5.1.1 Definition

The Kalma filter is a filtering technique commonly used for modelling dynamic models, such as GDP. It has several applications, including in the field of artificial neural networks.
Filter-Kalman is a signal processing algorithm used to estimate the hidden deviation of a dynamic system from noisy measurements. It is particularly effective for time series and when you want to estimate the future of the system. It operates in two main phases:

1. Prediction of future state
2. Update of the estimate using newly available measurements

Its strength is that it takes into account the uncertainty associated with measurements and predictions in order to produce an optimal estimate.
The Kalman filter is interesting in the case of GDP analysis because these data are often considered as a dynamic system. Indeed, GDP is a key measure of a country's economic performance and is influenced by many factors (discussed above). The aim of the filter is to model and predict fluctuations in GDP by taking into account the different variables that make up GDP.
We add a small illustration of how Filter-Kalman works:

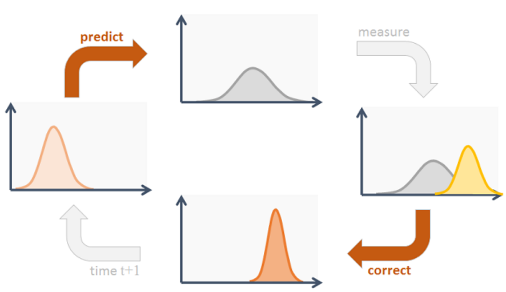


Figure 8: Illustration of Kalman-Filter

Finally, the Kalman filter is a filtering technique that makes it possible to use time series correctly and can be used in combination with ANN and RNN models to improve the quality of economic forecasts by pre-processing time series data. In the remainder of this report, we will develop the ANN model.

5.1.2 Implementations

In the Kalman filter framework, **'filtered_states'** represents the state estimates at each instant, given all observations up to that point. As mentioned previously, the Kalman filter is a recursive algorithm that estimates the states of a dynamic system in an optimal way by minimising the variance of the estimated error. It combines a series of observations measured over time, containing statistical noise and other uncertainties, with predictions of the future states of the system.

In our R code, after applying the **'dlmFilter'** to the time series data, **'filtered_states'** *[Appendix 5.1.2]* contains filtered estimates of GDP for each quarter. These estimates are obtained by taking into account both the model and the observed data up to each point in time.

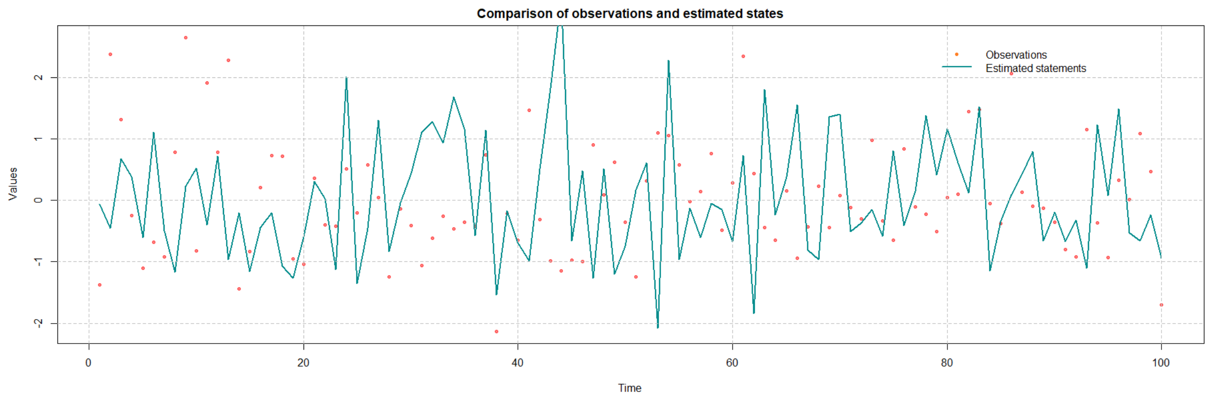


Figure 9: Comparison of observations and estimated states

This graph illustrates the application of a Kalman filter to Dutch log-GDP data. The observations are shown in red. These are real values measured in each period (quarter). Fluctuations between quarters reflect real changes in the economy, such as growth or recession.
In blue are the estimated states of the Kalman filter for GDP. It is used to determine the noise in the measurement and to provide a smooth estimate of the real trajectory. It can be interpreted as an underlying trend in GDP, cleaned of transitory irregularities.

In addition, we obtained a small **state_variance**, which suggests that the state latents have a small variance and therefore a good confidence in these Kalman-filter estimates.

```
> print(state_variances)
[1] 7.539262e-05
```

Moreover, we plot a graph of the residuals distribution:

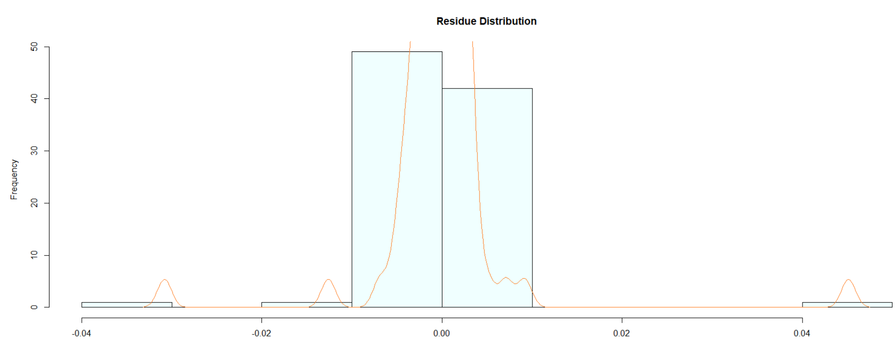


Figure 10: Residuals Distribution under Kalman-Filter

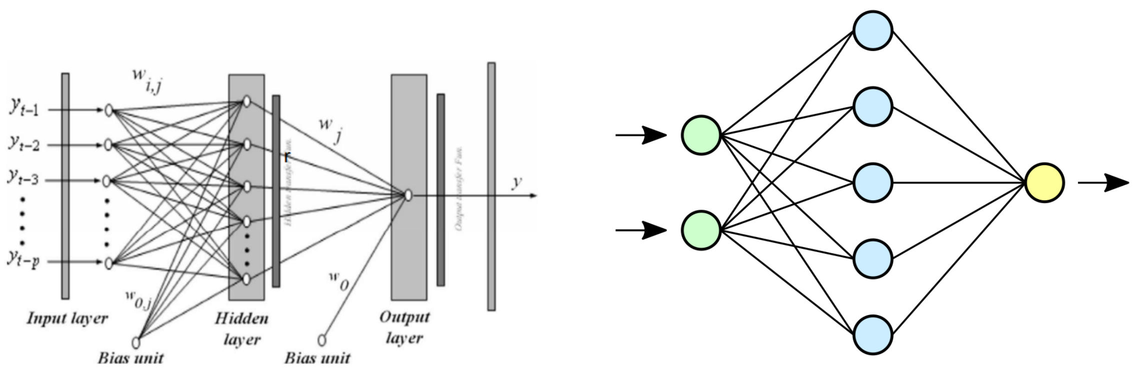
It is easy to see that most of the residuals are close to zero, which is a good sign! In fact, they are normally distributed and centred around zero with a constant variance. We can say that the model is effective in capturing the information in the data without leaving any systematic structure in the residuals.
As a reminder, the residuals represent the difference between the actual observations and the estimates generated by the Kalman-Filter in this case. They are therefore the model's prediction error at each time point.

5.2 Artificial Neural Networks (ANN's)

After using the Kalman-Filter (KF) model to model our time series and estimate the hidden state, taking into account the available observations and the associated noise, we turned to the integration of an ANN model. These two models are complementary and can be used for short-term prediction and anomaly detection. ANN is also useful for improving estimates of KF

5.2.1 Definition

Artificial neural networks offer an opportunity to model the non-linear relationship specific to economic data. In fact, this model can be used to model free dynamics. It can also model any form of unknown relationship in the data with some assumptions. According to Wikipedia, *an artificial neural network, or artificial neural network, is a system whose design was initially inspired schematically by the functioning of biological neurons, and which then moved closer to statistical methods.*



(a) Complexe ANN (b) Reduced ANN

Figure 11: Artificial neural network structure

ANNs are classroom models inspired by the workings of the human brain. They are used to solve a variety of automatic learning tasks. First, it is important to well understand the structure of ANN model.

The ANN is made up of three major layers, each of which is made up of neurons and two biasing units between them:

- **Input layer:** It is the starting point of an ANN and receives the input data which will be treated by the network. Each neuron of this layer represents a different input characteristic. This input layer distributes the data to the following layers without performing any calculation. The idea is simply to provide the input data to the neurons in the following layers
- **Hidden layer:** There are situated between the input layers and the output layers. There are called "hidden" as their output are not immediately visible. These layers are the heart of the processing in an ANN, they perform complex transformations on the inputs they receive from the previous layer using neurons with activation functions. The complexity of the model increases with the number of hidden layers and the number of neurons in each layer.
- **Output layer:** This is the last layer of an ANN and produces the final results of the network. This layer transforms the secret representations learned by the hidden layers into concrete, interpretable outputs.
- **Neurons:** Each neuron is a processing unit. In an ANN, it is a mathematical node that takes weighted inputs, performs a weighted sum and applies an activation function to produce an output. The neurons caclulate a weighted sum of the inputs such that : $\sum_{i=1}^{n} w_i x_i$

Architectures can vary from simple networks with a single hidden layer to deep networks with several layers. Each layer is made up of a number of neurons which receive input, process it and pass the output to the next layer. Most of the time, the choice of ANN structure is based on the nature of the data. And obviously, the performance of a model depends on its architecture. The more complex it is, the more it will be able to capture more complex relationships, but with a risk of over-fitting. So you need to choose the architecture carefully, depending on the size and complexity of the data.

5.2.2 Key Principles for the design and optimisation of ANN

- **Activation function:**
  Activation functions allow the neural network to capture non-linear relationships. Without these functions, the network would be limited to processing only linear functions. They therefore enable ANNs to model complex relationships. The most common fonction used is the ReLu (Rectified Linear Unit): $f(x) = \max(0, x)$, it is used to add non-linearity while remaining efficient.

- **Training:**
  Training an ANN involves adjusting the weights and biases to minimise a loss function that measures the error between the ANN's predictions and the actual values.

- **Loss function and Overfitting:**
  The loss function quantifies the difference between the ANN's predictions and the actual values. It determines how the ANN adjusts its weights during training.To measure them, it is common to use mean squared error (MSE):

  $$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{5.1}$$

  where y_i is the actual value and $\hat{y}_i$ is the prediction.

  Moreover, ANNs can be prone to over-fitting, where they adapt too well to training data but generalise poorly to new data. Techniques such as regularisation are used to avoid over-fitting.

- **Hperparameters:** Hyperparameters are model configuration parameters that are not modified during training. They include learning rate, batch size, number of learning epochs, etc. In general, they should be defined before training and are adjusted to optimise model performance. To find the best combination of hyperparameters, we can use the grid search method or the random search method, which are the most common methods.

We can also talk about advanced optimisation techniques which are very useful for improving the performance and efficiency of the learning process, and which are particularly important when working with large and complex datasets.

1. **Dropout:** It is a technique where neurons are randomly "deactivated" during training, which helps prevent overfitting.

2. **Batch Normalization:** This technique uses lds normalisation in each layer to improve the stability and speed of the drive.

3. **Residual Connections:** It helps to combat the problem of gradient disappearance in very deep networks by allowing activations to skip certain layers.

Finally, it is important to evaluate and validate the model. To do this, we can simply use the MSE or MAE (mean absolute error). Cross-validation is a robust technique for evaluating model performance on unseen data.

To conclude, an ANN is an informatic model inspired by the biological neural network of the human brain, used to recognise complex patterns through approximation functions. In the context of our work, the per-forecast analysis of Dutch GDP, an ANN can model the non-linear relationships between various economic divisors and GDP to improve the accuracy of economic forecasts.

5.2.3 Implementation

As we stated before, we first used the Kalman-Filter method as a pre-test to smoothened the GDP time series and use it properly afterwards. The purpose being to have more accurate and filtered values in the end to train our Artificial Neural network. To train our model, we decided to take a maximum lag of 4 to be used for the forecast and to train itself. Besides, we chose to assign 5 hidden layers to our model.The reason why we chose those parameters for the maximum lag and the hidden layers is to avoid over-specifying which, as mentioned above, can cause the model to overfit and therefore learn the model too well and include the noise in the forecast. As a consequence, the performance of our model could be diminished.

After the first step of initializing our model completed, we proceeded to the training of our ANN system. To this end, we computed a loop in order to predict the future values. Finally, one the loop was completed, we calculated the mean squared error to calculate the error metric. For such indicator describing accuracy, the closer to zero, the better it is. As a reminder, to return this indicator, we applied the formula (5.1). In our case, we obtained:

```
> MSE
[1] 1.185363e-05
```

which is a result close to zero encouraging us to trust our model.

Meanwhile, we also computed the 95% confidence interval to match the forecasted plots from the previous sections. Unfortunately, we obtained an expanding CI as the quarters grow which can be interpreted as heteroscedastic variables. Finally, we computed two plots showing the forecast of the ANN model for the 3 upcoming years.

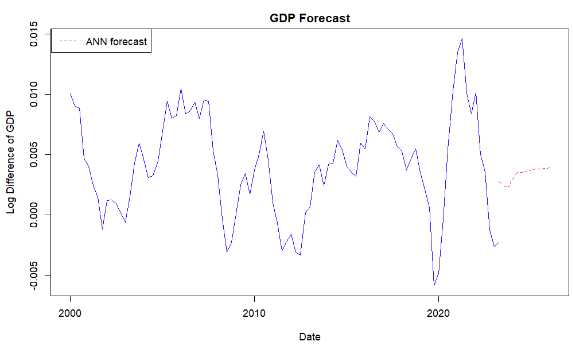


Figure 12: Dutch GDP growth with ANN forecast

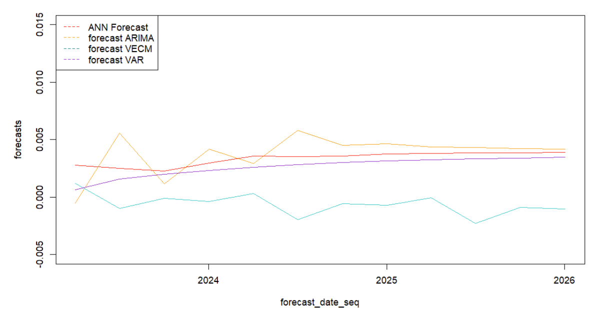


Figure 13: Comparison of all the modeled forecasts

The first one Figure (12) shows the Dutch's GDP growth with the ANN forecast. The second one, Figure (13) helps us visualize and compare the different forecasts computed along this project. Visually, we can conclude that the ANN and VAR forecast are really close to one another and almost follow the same predicted path. Although we can observe that the ARIMA's is slightly deviating, it converges toward the same final predict value of the real GDP.

On the other hand, the VECM forecast appears rather singular when compared to the others. Indeed, it fluctuates more and tends to be less optimistic concerning the future values taken by our time series. This can be explained by the differences between the models. Indeed the VECM includes co-integration inside its model and is originally designed for non-stationary time series. The inclusion of an error corrected term might be the reason why it deviates, specifically when keeping in mind that our time series were stationary. The lag operator included in the model could also have been a cause, but the one chosen was the one resulting from the best MSE value so we can exclude this assumption.

On the bright side, even though we may have doubts about our VECM relationship with the time series, we observe that the new model tested, ANN, thanks to the kalman Filter observed a path similar to other proven forecasting methods.

In *Appendix 5.2.3. (17)*, we sketched an Artificial Neural Network to represent the actual structure of the model with 4 lags as input. We can also observe the 5 hidden layers in the model converging in the end toward the final targeted value. The different arrows represent the relationship, also called weight of each nodes on the next. Concerning the blue nodes and arrows, they represent the bias. They show how the node's output is adjusted. Those bias are indeed connected to the neural network and act as intercept. They are understood to fit the data better and add some flexibility to the model i.e. by shifting to the left or the right the activation function. This appendix has been mainly added to have a more visual approach of what the model looks like.

5.3 Impact of the Covid-19 Pandemic on Dutch GDP Forecasts

As final task, we wanted to compare the different forecasts that we have been able to compute all along this project. As a visual reminder, you can see figure 14 that represents the different path following the different methods. In this section, we will also explore how the Covid-19 pandemic can affect our different forecasting models.

As we have already stated many times, one of the best ways to check the accuracy of any time series with regard to an original one, is to match the predicted value by a model ("fitted values") to the actual ones. Here again, we will use the MSE and even the RMSE to evaluate the different fitted values of the ARIMA, VAR, VECM and Kalman Filter methods. As a reminder, the RMSE is equal to the squared root of the MSE and is preferred to the latter.

We first had to clarify what is included in the "Covid period" considered as outliers in our computation and that we will therefore ignore. This period will be used to divide every fitted time series into two distinct ones: "before the pandemic" and "after the pandemic". With the help of the outliers from the first part, we decided that we would ignore the metrics from the year 2020 represented by quarters 80 to 84 (81-85 for the Kalman-Filter). As a result, we computed and compared the different RMSE. On the left, we have the values before the pandemic and on the right is after 2020:

```
rmse_arimaB: 0.003988466
rmse_varB: 0.002598862
rmse_vecmB: 0.003664841
rmse_KFB: 0.003664841
```

```
rmse_arimaA: 0.006696662
rmse_varA: 0.004987174
rmse_vecmA: 0.006947937
rmse_KFA: 0.003035893
```

When looking at the observed values, we can deduct that for almost all the methods, the RMSE is higher for the period after the covid crisis than before, except for the Kalman-Filter model for which it is the other way around. The reason why the RMSE gives us less confidence in the fitted values after the covid can be explained by the fewer data available.

On a more economical perspective, it may be explained by the uncertainty and uneasiness of markets. As a matter of fact, the Dutch real GDP, as any other depends on a variety of factors which are mainly markets. Since the pandemic has changed our consumption and views on economical market, it may have a direct effect on the ability of the model to predict accurate data. Even if the RMSE doubled from one period to another, it is important to pinpoint that the values are still close to zero.

Finally, below is a graph showing all the adjusted values according to their methods:

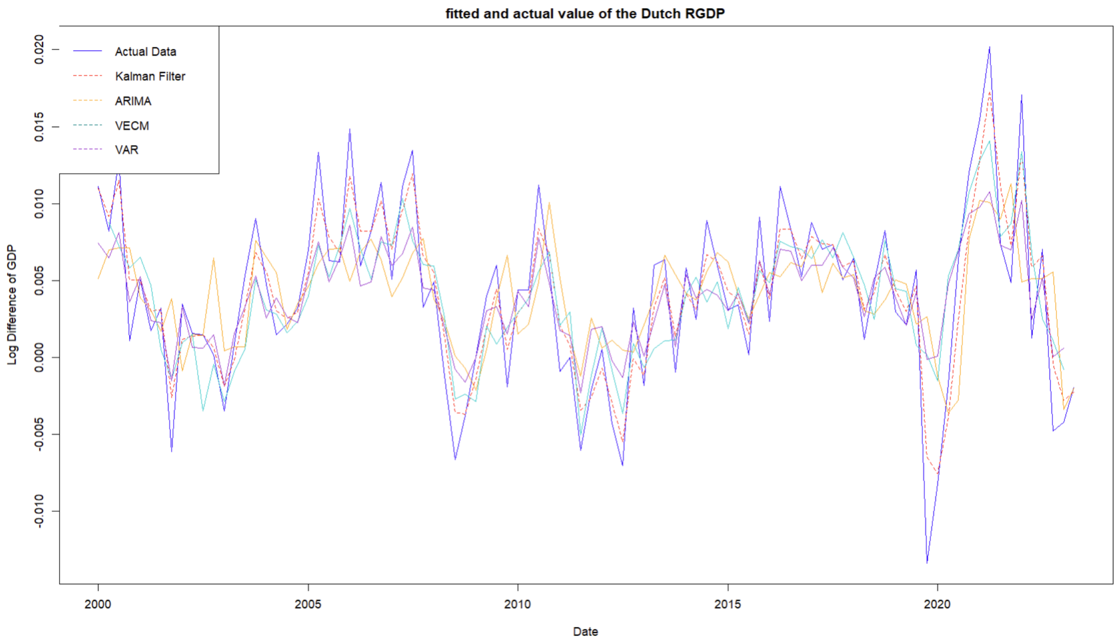


Figure 14

This plot actually confirms our assumption regarding which the different models represent well the actual values take by the difference of logarithm of the Dutch's GDP. Again, we can visually observe that they all follow a similar path for the fitted values. It means that the patterns fluctuate but not exactly the same way or at the same scale as the others.

6 Conclusion

6.1 Limits

Throughout this work we have come to realise that there are certain limitations to this analysis.

Firstly, as we know, econometric models are based on assumptions that may not always prove to be true, or that may change according to market conditions or unforeseen events. Consequently, the assumed linearity of the relationships between the variables and the stationarity of the time series may not faithfully reflect real economic behaviour.

Then, given that we have chosen to take our data from 2000 onwards, they are limited historically. This raises the question of the applicability of the results to future periods. However, we were careful to take a fairly wide range of years so as not to have too many problems.

In addition, although we have taken care to ensure that the variables are significant, the selection of variables we have made is still open to debate. Indeed, there may be a risk of over-fitting or, conversely, of missing important nuances if we had chosen too little data, leading to inaccurate forecasts.

Finally, the forecasts generated by the various models do not take into account future economic policy decisions, which can have a major impact on economic growth. Our forecasts must therefore be updated regularly with new data and in the context of the latest economic and political developments.

6.2 Conclusion

Throughout this report, we have analysed GDP in the Netherlands since 2000 and for the next three years. We have used statistical methods as well as econometric models such as the ARIMA/ARIMAX models. The latter enabled us to identify significant trends and exogenous indicators that have an influence on GDP.
We were therefore able to enrich our model by adding variables such as energy prices, economic policy and exchange rates, for example.

We then carried out a multivariate analysis using VAR and VECM models, taking into account various economic indicators. These models revealed the importance of employment, production and trade in influencing the GDP of the Netherlands. Using Granger-Causal tests and cointegration, we were able to validate the relevance of these latter variables.

Finally, we tackled the innovation part, with the integration of Kalman-Filter and Artificial Neuron Networks. These new techniques have enabled us to refine our forecasts by taking into account the uncertainties and complex dynamics of the economic data.

In conclusion, although our models obviously cannot predict with certainty the future evolution of GDP in the Netherlands, they do offer an insight into potential future trends. The forecasts suggest weak but steady GDP growth over the next three years, marked by a cautious economic recovery from the Covid-19 global crisis. This report therefore provides a solid basis for political and economic decision-making, underlining the importance of ongoing analysis and possible adaptation to changing market conditions.

7 Appendices

Appendix 3.4

```
Series: gdp_clean
ARIMA(2,0,0)(0,0,2)[4] with non-zero mean

Coefficients:
         ar1     ar2      sma1      sma2    mean
      0.3527  0.2618   -0.3287   -0.2045   4e-03
s.e.  0.1012  0.1030    0.1144    0.1125   6e-04

sigma^2 = 2.379e-05:  log likelihood = 368.94
AIC=-725.88   AICc=-724.91   BIC=-710.62
```

Appendix 3.5

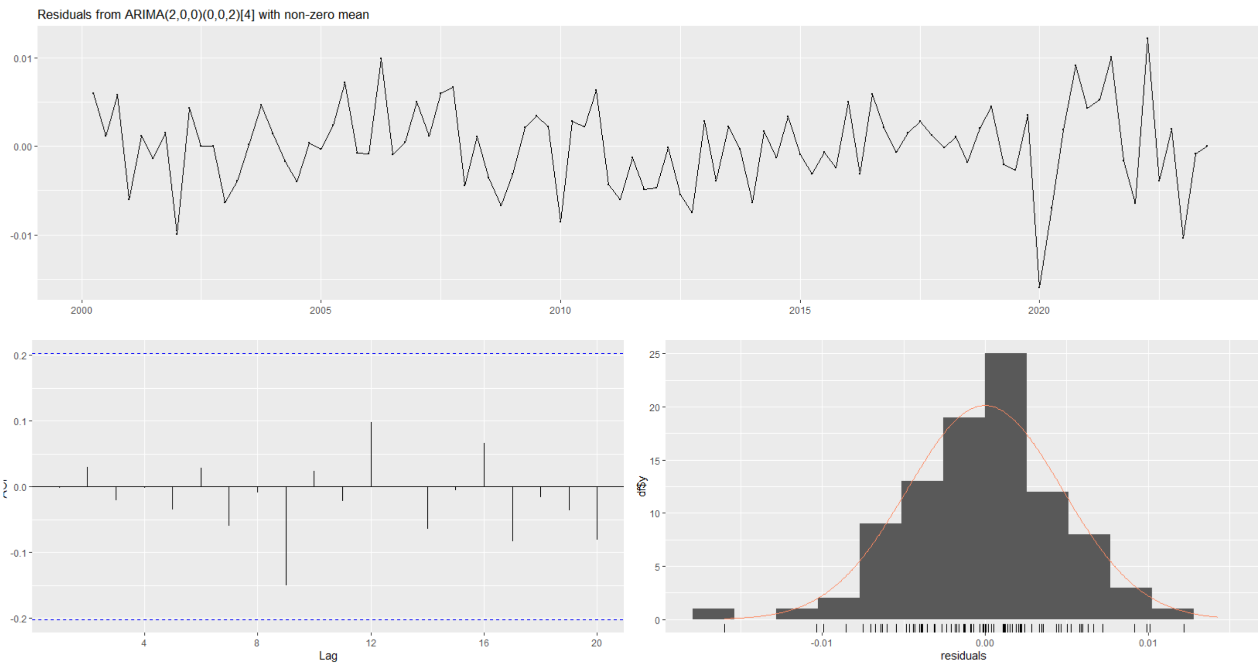


Figure 15: CheckResiduals

Appendix 3.6

Quarter	Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
2023 Q4	-0.0005375168	-0.006787633	0.005712599	-0.010096244	0.009021211
2024 Q1	0.0055682197	-0.001059307	0.012195747	-0.004567708	0.015704148
2024 Q2	0.0011553849	-0.005898084	0.008208854	-0.009631965	0.011942735
2024 Q3	0.0041850812	-0.003011620	0.011381782	-0.006821323	0.015191485
2024 Q4	0.0029340188	-0.004321053	0.010189091	-0.008161656	0.014029694
2025 Q1	0.0057938227	-0.001461422	0.013049067	-0.005302117	0.016889762
2025 Q2	0.0045293098	-0.002729355	0.011787975	-0.006571860	0.015630480
2025 Q3	0.0046497939	-0.002609166	0.011908754	-0.006451828	0.015751416
2025 Q4	0.0043723636	-0.003012843	0.011757570	-0.006922335	0.015667062
2026 Q1	0.0043060511	-0.003095842	0.011707945	-0.007014168	0.015626270
2026 Q2	0.0042100258	-0.003210904	0.011630956	-0.007139307	0.015559359
2026 Q3	0.0041587936	-0.003268922	0.011586509	-0.007200917	0.015518505

Table 1: Model previsions

Appendix 3.8

```
> ARIMAX_1 <- arima(gdp2k_clean, order = c(2,0,0), xreg = Prod2K_ld)
> summary(ARIMAX_1)

Call:
arima(x = gdp2k_clean, order = c(2, 0, 0), xreg = Prod2K_ld)

Coefficients:
         ar1     ar2  intercept  Prod2K_ld
      0.3397  0.2075     0.0039      8e-04
s.e.  0.1035  0.1018     0.0011      2e-04

sigma^2 estimated as 2.205e-05:  log likelihood = 370.42,  aic = -730.85

Training set error measures:
                        ME        RMSE         MAE       MPE     MAPE     MASE        ACF1
Training set -5.672275e-05 0.004695408 0.003509364 -1217.641 1423.837 0.716518 0.009108866
```

```
> ARIMAX_2 <- arima(gdp2k_clean, order = c(2,0,0), xreg = unemp2K_ld)
> summary(ARIMAX_2)

Call:
arima(x = gdp2k_clean, order = c(2, 0, 0), xreg = unemp2K_ld)

Coefficients:
         ar1     ar2  intercept  unemp2K_ld
      0.2945  0.1018      4e-03     -0.0268
s.e.  0.1127  0.1121      8e-04      0.0122

sigma^2 estimated as 2.452e-05:  log likelihood = 365.51,  aic = -721.02

Training set error measures:
                       ME        RMSE         MAE       MPE     MAPE      MASE         ACF1
Training set -3.55013e-05 0.004951485 0.003763179 -239.7394 485.2739 0.7683402 -0.002795472
```

```
    > ARIMAX_3 <- arima(gdp2k_clean, order = c(2,0,0), xreg = edh2K_ld)
> summary(ARIMAX_3)

Call:
arima(x = gdp2k_clean, order = c(2, 0, 0), xreg = edh2K_ld)

Coefficients:
         ar1     ar2  intercept  edh2K_ld
      0.3817  0.1537     0.0041   -0.0144
s.e.  0.1021  0.1056     0.0011    0.0286

sigma^2 estimated as 2.577e-05:  log likelihood = 363.1,  aic = -716.2

Training set error measures:
                        ME        RMSE         MAE       MPE     MAPE      MASE        ACF1
Training set -6.116892e-05 0.005076221 0.004000226 -613.5063 793.5279 0.8167388 0.006860331
```

```
> ARIMAX_3 <- arima(gdp2k_clean, order = c(2,0,0), xreg = exp2K_ld)
> summary(ARIMAX_3)

Call:
arima(x = gdp2k_clean, order = c(2, 0, 0), xreg = exp2K_ld)

Coefficients:
         ar1     ar2  intercept  exp2K_ld
      0.2729  0.2761     0.0035    0.0326
s.e.  0.1027  0.1011     0.0011    0.0089

sigma^2 estimated as 2.277e-05:  log likelihood = 368.89,  aic = -727.78

Training set error measures:
                        ME        RMSE         MAE       MPE     MAPE      MASE       ACF1
Training set -7.243787e-05 0.004772043 0.003740321 -515.2668 640.2614 0.7636732 0.01637531
```

```
> ARIMAX_4 <- arima(gdp2k_clean, order = c(2,0,0), xreg = imp2k_ld)
> summary(ARIMAX_4)

Call:
arima(x = gdp2k_clean, order = c(2, 0, 0), xreg = imp2k_ld)

Coefficients:
         ar1     ar2  intercept  imp2k_ld
      0.2758  0.2547     0.0036    0.0335
s.e.  0.1028  0.1012     0.0010    0.0089

sigma^2 estimated as 2.26e-05:  log likelihood = 369.27,  aic = -728.54

Training set error measures:
                        ME        RMSE         MAE       MPE     MAPE      MASE        ACF1
Training set -7.23783e-05 0.004753602 0.003760645 -571.8074 697.8293 0.7678228 0.006854855
```

Appendix 4.1.1

```
Call:
lm(formula = gdp2k_ld ~ unemp2K_ld + exp2K_ld + imp2k_ld + govexp2K_ld +
    SER2K_ld + Prod2K_ld + EPU2K_ld + REER2K_ld + EP2K_ld + RM2K_ld +
    edh2K_ld + oil2k_ld + PPI2K_ld + ret2k_ld)

Residuals:
      Min        1Q    Median        3Q       Max
-0.035305 -0.003558  0.000175  0.004821  0.023082

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.0004675  0.0013923   0.336 0.737905
unemp2K_ld  -0.0372387  0.0164929  -2.258 0.026714 *
exp2K_ld     0.2307605  0.0657886   3.508 0.000749 ***
imp2k_ld    -0.0479283  0.0590943  -0.811 0.419776
govexp2K_ld  0.0916116  0.0764122   1.199 0.234145
SER2K_ld    -0.1120640  0.0719491  -1.558 0.123338
Prod2K_ld    0.0010557  0.0005094   2.072 0.041484 *
EPU2K_ld    -0.0015362  0.0042960  -0.358 0.721604
REER2K_ld   -0.0187122  0.0912653  -0.205 0.838076
EP2K_ld      0.0001059  0.0193390   0.005 0.995643
RM2K_ld     -0.0421689  0.0197463  -2.136 0.035815 *
edh2K_ld    -0.0437884  0.0534335  -0.819 0.414970
oil2k_ld     0.0061452  0.0154246   0.398 0.691406
PPI2K_ld    -0.1440122  0.0538318  -2.675 0.009075 **
ret2k_ld     0.1630675  0.0558478   2.920 0.004562 **
---
Signif. codes:  0    ***    0.001    **    0.01    *    0.05    .    0.1         1

Residual standard error: 0.008329 on 79 degrees of freedom
Multiple R-squared:  0.6727,  Adjusted R-squared:  0.6148
F-statistic:  11.6 on 14 and 79 DF,  p-value: 6.005e-14
```

Appendix 4.1.2

```
Call:
lm(formula = gdp2k_ld ~ PPI2K_ld + exp2K_ld + ret2k_ld + unemp2K_ld +
    Prod2K_ld + RM2K_ld + REER2K_ld + SER2K_ld + PCE2K_ld)

Residuals:
      Min        1Q    Median        3Q       Max
-0.036271 -0.003225  0.000066  0.003970  0.025345

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.0014435  0.0009532   1.514 0.133699
PPI2K_ld    -0.1697483  0.0502834  -3.376 0.001116 **
exp2K_ld     0.1405953  0.0383155   3.669 0.000426 ***
ret2k_ld     0.1782119  0.0496476   3.590 0.000556 ***
unemp2K_ld  -0.0400969  0.0149315  -2.685 0.008728 **
Prod2K_ld    0.0011078  0.0004816   2.300 0.023912 *
```

```
RM2K_ld      -0.0481236  0.0183616  -2.621 0.010408 *
REER2K_ld    -0.0526758  0.0842455  -0.625 0.533492
SER2K_ld     -0.0658325  0.0633315  -1.039 0.301560
PCE2K_ld      0.0586667  0.0218797   2.681 0.008826 **
---
Signif. codes:  0    ***    0.001    **    0.01    *    0.05    .    0.1         1

Residual standard error: 0.007965 on 84 degrees of freedom
Multiple R-squared:  0.6818,  Adjusted R-squared:  0.6477
F-statistic:    20 on 9 and 84 DF,  p-value: < 2.2e-16
```

Appendix 4.2.3

```
    Granger causality H0: gdp2k_clean do not Granger-cause Prod2K_ld UE_clean
  PCE2K_clean

data:  VAR object Varm
F-Test = 1.6084, df1 = 3, df2 = 352, p-value = 0.1871


  Granger causality H0: UE_clean do not Granger-cause gdp2k_clean Prod2K_ld
  PCE2K_clean

data:  VAR object Varm
F-Test = 3.9427, df1 = 3, df2 = 352, p-value = 0.008682

  Granger causality H0: Prod2K_ld do not Granger-cause gdp2k_clean UE_clean
  PCE2K_clean

data:  VAR object Varm
F-Test = 0.42477, df1 = 3, df2 = 352, p-value = 0.7354


  Granger causality H0: PCE2K_clean do not Granger-cause gdp2k_clean Prod2K_ld
  UE_clean

data:  VAR object Varm
F-Test = 2.2641, df1 = 3, df2 = 352, p-value = 0.08077
```

Appendix 4.3.

```
Coefficient matrix of lagged endogenous variables:

A1:
            gdp2k_clean.l1  Prod2K_ld.l1 UE_clean.l1 PCE2K_clean.l1
gdp2k_clean      0.2993329  0.0004021663 -0.01175581  -0.0009147058
Prod2K_ld       68.3970194 -0.2200460723  9.23744636  10.0647555311
UE_clean        -0.9406695 -0.0017907339  0.71356509   0.0672201936
PCE2K_clean      0.7183295  0.0010688778 -0.40467779   0.1647531100


A2:
            gdp2k_clean.l2  Prod2K_ld.l2 UE_clean.l2 PCE2K_clean.l2
gdp2k_clean     0.09721637  0.0001285082  -0.0345341    0.006200861
Prod2K_ld     -29.93545837 -0.1062813692 -11.4148847    2.366613595
UE_clean       -1.01475854 -0.0007533205   0.1206706   -0.002178595
PCE2K_clean    -1.12080104  0.0024624884   0.2198959   -0.031945311


Coefficient matrix of deterministic regressor(s).

                constant          sd1          sd2          sd3
gdp2k_clean  0.001720236  0.001952597  0.001685576  0.002515995
Prod2K_ld   -0.262255269 -0.194836190 -0.958724888 -0.292251091
UE_clean     0.006739237  0.009158618  0.010379432  0.004005580
PCE2K_clean  0.012409566  0.018346136  0.017237537  0.001874034
```

Appendix 4.3.1

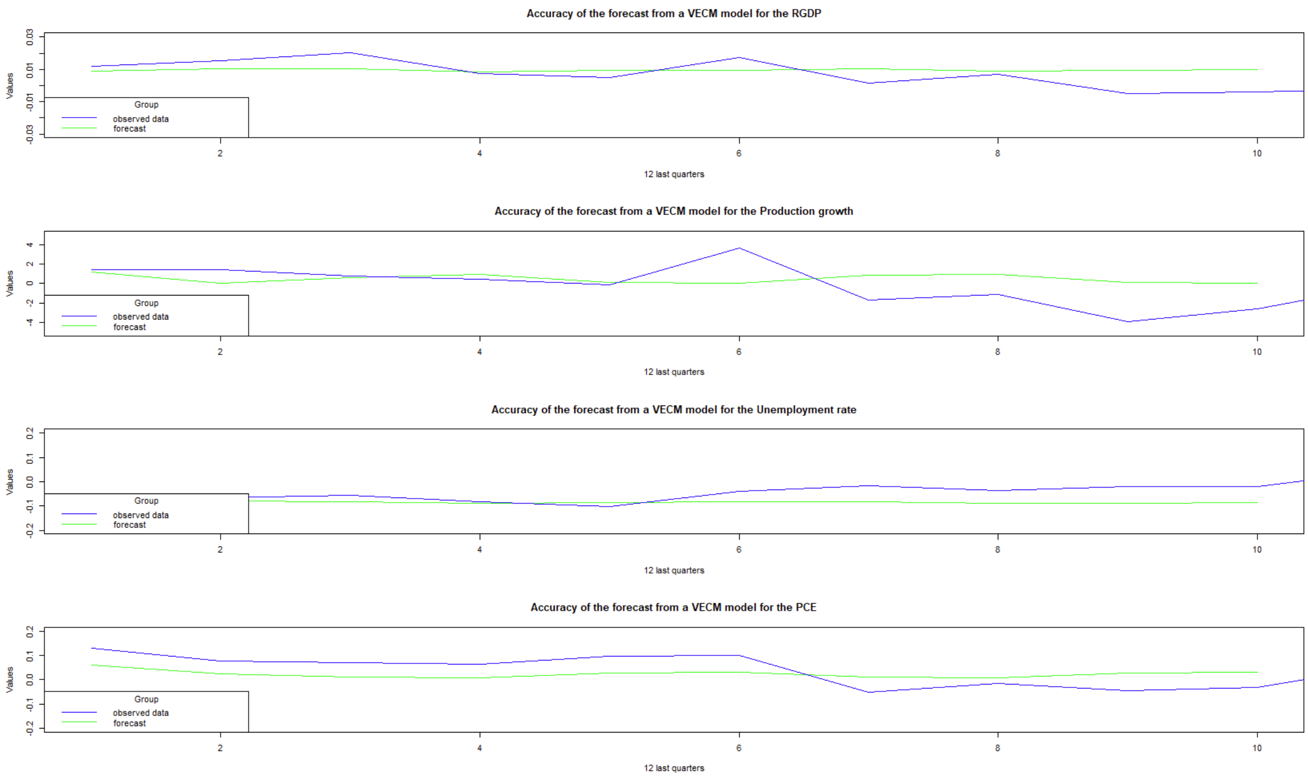


Figure 16: Accuracy of the forecasted VECM model

Appendix 5.1.2

Year	Qtr1	Qtr2	Qtr3	Qtr4
2000	0.0000000000	0.0109975570	0.0091471023	0.0115005626
2001	0.0050529631	0.0050472908	0.0030010186	0.0031363411
2002	-0.0025848107	0.0011614416	0.0014026601	0.0014597235
2003	0.0006647816	-0.0018840183	-0.0001716096	0.0032347408
2004	0.0068049858	0.0055389332	0.0030382072	0.0025066795
2005	0.0029496062	0.0054589860	0.0103297522	0.0078306993
2006	0.0068351899	0.0117942984	0.0081906914	0.0082033881
2007	0.0101741177	0.0070401884	0.0095572030	0.0119566802
2008	0.0065880605	0.0055531004	0.0013656770	-0.0035704643
2009	-0.0239826074	-0.0091448030	-0.0010216264	0.0033213907
2010	0.0001001458	0.0027450337	0.0037732758	0.0083747756
2011	0.0067553598	0.0020274402	0.0007770780	-0.0034435830
2012	-0.0026031262	-0.0006796813	-0.0029178531	-0.0054544732
2013	-0.0000909781	-0.0011525798	0.0032754481	0.0051616043
2014	0.0013850453	0.0041369025	0.0031246253	0.0067043053
2015	0.0062269118	0.0042800689	0.0037645215	0.0015371849
2016	0.0062308848	0.0038377557	0.0083380913	0.0083379885
2017	0.0064197903	0.0078720352	0.0073683046	0.0073429539
2018	0.0059232654	0.0062750840	0.0031275092	0.0041997841
2019	0.0066987017	0.0043944015	0.0029886460	0.0046630528
2020	-0.0064696275	-0.0563343600	0.0166935945	0.0103037992
2021	0.0113708398	0.0228562672	0.0212025627	0.0126560553
2022	0.0078313244	0.0135359270	0.0059471812	0.0066138089
2023	-0.0004229886	-0.0027648569	-0.0022490070	...

Table 2: Filtered States

Appendix 5.2.3

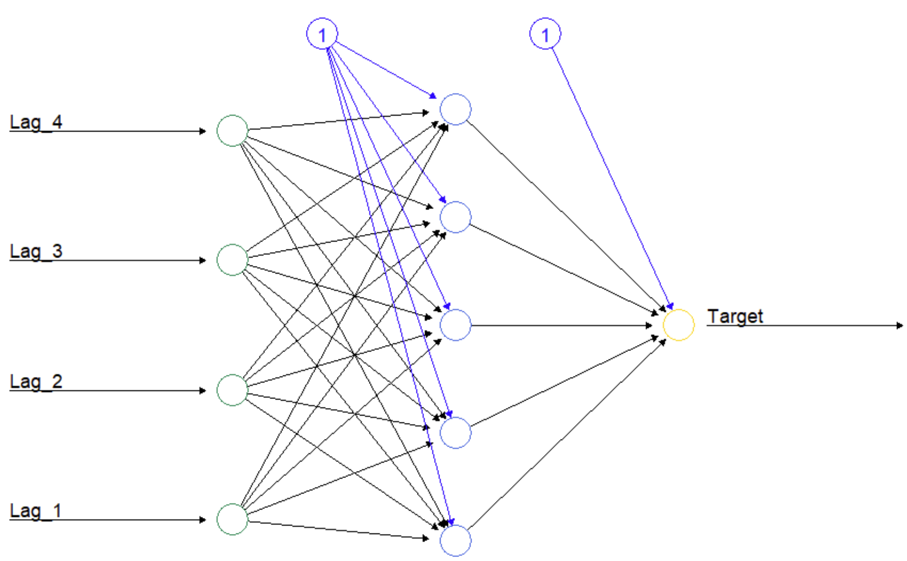


Figure 17: Artificial Neural Netword modeled for the GDP

8 Bibliography

References

[1] Tusell, F. (2011). Kalman filtering in R. Journal of Statistical Software, 39, 1-27.

[2] Anindita, S. (2022, January 6). Artificial Neural Network using R Studio - SUKMA ANINDITA - Medium. Medium. https://medium.com/@sukmaanindita/artificial-neural-network-using-r-studio-3eb538fa39fb

[3] LLSM224- Forecasting-23/24, Lesson 6 and 7. (n.d.). https://moodle.uclouvain.be/pluginfile.php/552698/mod_resource/content/1/Lecture%207.pdf

[4] Wikipedia contributors. (2024, January 2). Artificial neural network. Wikipedia. https://en.wikipedia.org/wiki/Artificial_neural_network

[5] Hubbard, A. (2023, September 25). Kalman filter for state space models. https://cran.r-project.org/web/packages/kalmanfilter/vignettes/kalmanfilter_vignette.html

[6] Schmelzer, C. H. M. a. a. G. a. M. (2023, November 8). Introduction to Econometrics with R. https://www.econometrics-with-r.org/

[7] Federal Reserve Economic Data | FRED | St. Louis Fed. (n.d.). https://fred.stlouisfed.org/

[8] Singh, G. (2023, July 18). Introduction to artificial neural networks. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/09/introduction-to-artificial-neural-networks/

[9] Zhang, G., Patuwo, B. E., Hu, M. Y. (1998). Forecasting with artificial neural networks: International Journal of Forecasting, 14(1), 35–62. https://doi.org/10.1016/s0169-2070(97)00044-7