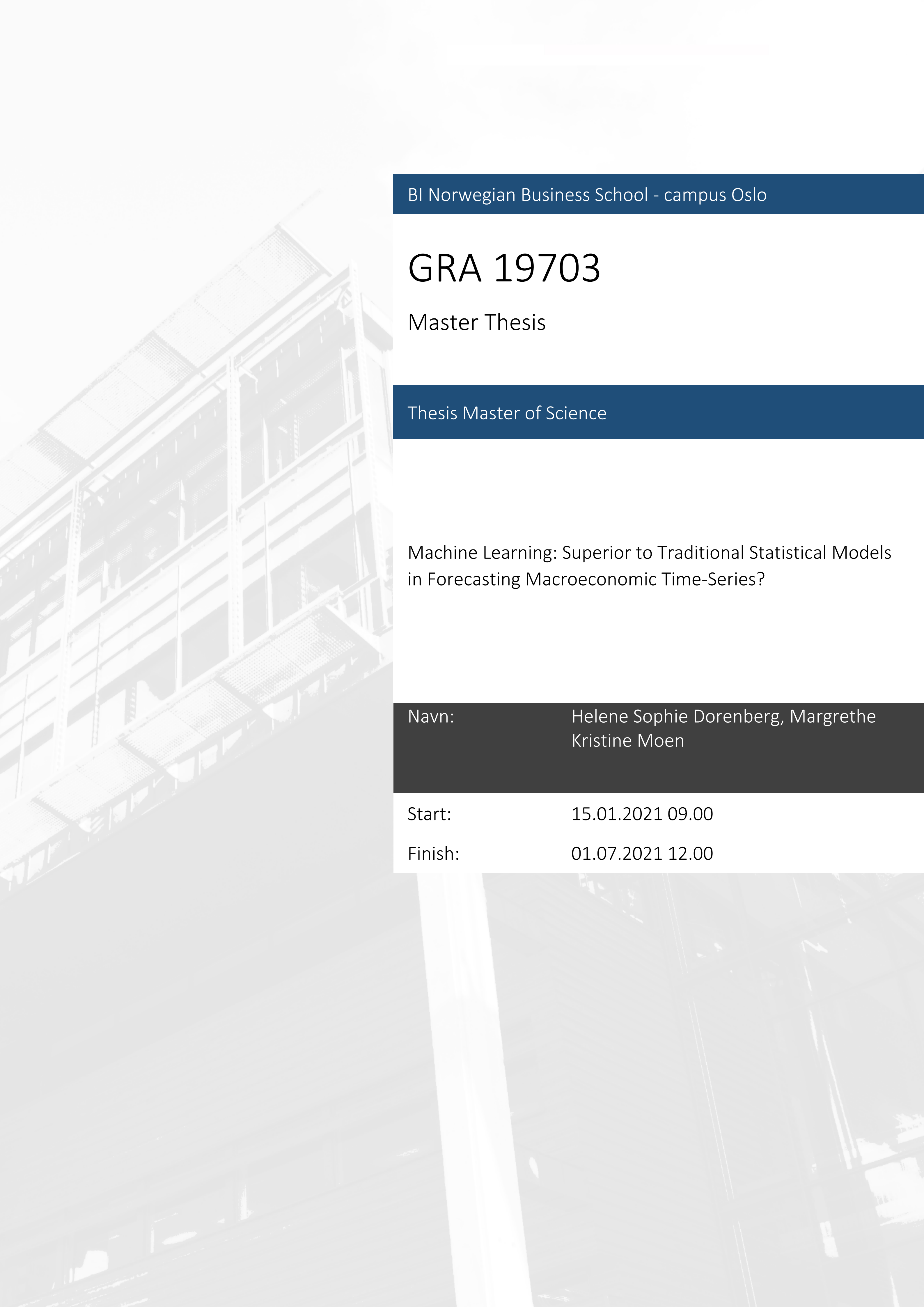


BI Norwegian Business School - campus Oslo

GRA 19703

Master Thesis

Thesis Master of Science

Machine Learning: Superior to Traditional Statistical Models in Forecasting Macroeconomic Time-Series?

Navn:	Helene Sophie Dorenberg, Margrethe Kristine Moen
Start:	15.01.2021 09.00
Finish:	01.07.2021 12.00

Machine Learning: Superior to Traditional Statistical Models in Forecasting Macroeconomic Time–Series?

Master Thesis

By
Helene Sophie Dorenberg
Margrethe Kristine Moen
MSc in Quantitative Finance

Supervisor
Costas Xiouros

Oslo, June 30, 2021

ABSTRACT

This thesis investigates machine learning's potential to forecast the Norwegian GDP, unemployment rate, and inflation on monthly or quarterly, and annual terms. We compare machine learning techniques such as penalised regressions and random forest to traditional statistical methods such as the naïve model, autoregressive and vector autoregressive models. This motivates the following thesis question, *Is value added by machine learning compared to traditional statistical models in time–series forecasting of macroeconomic variables?* The results show that the machine learning models are relatively better than the traditional statistical models when forecasting except for inflation. Using many exogenous variables to explain inflation is more confusing than value–adding, therefore, the models depending only on inflation itself provide the best forecasts.

This thesis is a part of the MSc programme at BI Norwegian Business School. The school takes no responsibility for the methods used, results found, or conclusions drawn.

Acknowledgements

We want to thank our thesis supervisor Professor Costas Xiouros for his guidance and detailed feedback when writing this thesis. In addition, we wish to thank Dane Cekov at Nordea Markets Large Corporates & Institutions for help with downloading the variables we needed through Macrobond.

Contents

List of Figures

List of Tables

1 Introduction and Motivation

Changes in macroeconomic factors provide an overall view of society's economy and are of interest both for individuals and larger corporations. Having a model that provides an accurate forecast is very valuable for financial institutions. Inflation, Gross Domestic Product (GDP), and unemployment rate are all giving strong indications about how the economy in a country is developing, which can affect strategies and the institutions' economic outlook. During recessions, people lose their jobs, and companies do not start employing until the economy expands again. Therefore, we look at the unemployment rate as a lagged variable (Amadeo (2020)). Establishing a layoff plan is time–consuming, so often it instead says something about the effect of events and indicates when the economy has recovered after a crisis. An expanding economy in terms of GDP combined with a declining unemployment rate is a clear sign that the economy is performing well. Regarding inflation, most central banks tend to adjust the interest rates to keep the inflation at a stable level, to prevent the economy from growing or dropping too fast (Picardo (2020)). Understanding the country's current and future economic situation helps maintain stability and enables the country to be better prepared and equipped during recessions. Norwegian GDP, unemployment rate, and inflation are our endogenous variables in the models as we believe prediction with high accuracy of these three factors in the short and long term is of great advantage. Understanding these concepts and how essential it is to keep up with the times motivates us to create machine learning models to see the value–added compared to simpler models like naïve forecasts and traditional autoregressive models (AR). We use random forest, elastic net, lasso and ridge regressions, in addition to a model that averages the results of those four models.

The world is constantly becoming more data–oriented, which has resulted in an extended interest in developing models and methods for specific purposes.

There are no longer just scientists who have their eyes on complex models. Financial policymakers also see great value in stepping away from leaning solely on behavioural finance and traditional econometrics tools, and many institutes find themselves moving into the realm of big data. Machine learning opens a world of valuable models that take all available data into account, finding connections and trends that are not obvious for economists.

During our studies we have focused on quantitative methods in finance and are familiar with processing financial data in conjunction with machine learning. This thesis addresses macroeconomic variables that handle two dimensions of data, cross–sectional and time–series data. The latter we are least familiar with and wish to study further the use of machine learning with panel data. We find this to be an exciting topic where we could apply previous and develop new knowledge. Even though several studies are done on time–series in machine learning, it is not as common for macroeconomic purposes. Few studies look at Norway's economy, extending the studies made on GDP as the master thesis by Bankson and Holm (2019) did, to now including several macroeconomic factors.

Time and knowledge are natural limitations. There are several machine learning models with different modifications, and to explore every single variant would be too extensive for our thesis. Therefore, it is important to note that the results can be improved by using other methods within machine learning and even more complex statistical methods. Data leak is a common phenomenon in time–series forecasting. It addresses the possibilities that models can be trained on observations that are not available or known in real–time. There is room for data leak and missing values due to large data sets, and this is in focus. However, the occurrence of data leakage cannot be ruled out.

Machine Learning can also be sensitive to how we part our data into training and test sets. With time–series data, we cannot mix the observations randomly into two sets as we must keep in mind the chronological order of observations. We apply a “rolling forward method” that extends the train set in each run to validate our model. Another limitation is the trade–off between the number of observations, n versus predictors, p. Some significant variable predictors have shorter time samples, and we create different data sets to run our model to see which composition gives the best forecasts.

2 Literature Review

Machine Learning is a part of artificial intelligence that through experience learn and improve the models without being explicitly programmed. As a technique it has several strengths and limitations. A machine learning model can quickly, with an extensive data set, identify hidden patterns and trends that the human eye cannot catch (DataFlair (2018)). Machine learning can learn, thus the name, meaning it does not require human intervention every step of the way. Through gaining experience, the algorithms develop continuous improvement as it improves accuracy. Additionally, machine learning is excellent at handling large data sets with many observations, and it has broad applications for both time–series forecasting and classification problems.

Even though machine learning has several strengths, there are some limitations as well. The models often require big data, and at times, data collection can be problematic. The most severe limitation of machine learning is interpretability. Most old–fashioned firms use only traditional statistical methods because they value interpretability. Making customers trust outcomes from a model where it is difficult to understand the underlying concepts or how the explanatory variables are put together can be challenging, making machine learning less attractive to use in practice (Stewart (2020)). Despite the limitations, we are interested in figuring out the additional value of machine learning compared to traditional tools and see if the value–added exceeds the disadvantages.

Many see great value in using machine learning for forecasting purposes. Studies show the importance of applying non–linear models to financial and macroeconomic data and how a program can process more information than economic and financial agents. Financial data is rarely normally distributed nor has a linear relation; hence we expect non–linear models to outperform linear ones. Traditional forecasting models often search for a fitted function to a pre–specified relationship between the response and explanatory variables.

Recently, researchers have wished to step away from any pre–specifications by using more data–driven methods to find the true relationship between the variables.

Financial institutes often apply surveys to help predict different outcomes of macroeconomic factors. Bianchi et al. (2020) use machine learning to estimate expectational errors embedded in survey responses. They discuss how the forecast of macroeconomic factors oscillates between optimism and pessimism according to changes in the total economic activity. Moreover, they study the impact of economists' beliefs on expectational errors and how they can be discovered using machine learning. Macroeconomic factors, e.g., the unemployment rate, can be vulnerable when a unique situation occurs, such as the financial crisis in 2008 and the global pandemic Covid–19. Expectational errors made by economic agents could be over– or underreacting to incoming news, holding onto outdated conservative models, motivating the implementation of machine learning. However, since it is rational that thoughts on macroeconomic variables can be emphasised on surveys, Bianchi et al. also discuss the possibility that some of the information may have been unavailable to survey respondents. Further, they emphasise the importance of avoiding data leakage in the machine learning models, making sure that no predictions at time t are based on information from time $t+1$.

Chakraborty and Joseph (2017) published a working paper for the Bank of London on how machine learning can contribute to value creation for central banks. Machine learning is often considered to have a "black–box nature" as the application of machine learning leads to a loss of interpretation of the variables' relationships. That is problematic for central banks as they rely on making informed decisions. However, their paper shows that there are ways to interpret the models, for instance, by limiting the model's complexity or work on smaller models. In fear of lack of interpretation, the elastic net model can be considered the most consistent since it originates from Ordinary Least Squares (OLS) and interprets the contribution of each variable when building the model (Jung et al. (2018)). As opposed to central banks, financial institutions have more freedom to rely "blindly" on outputs from a machine learning model if it yields good predictions, thus interpretation is less critical. This thesis weights the improvement of predictions rather than understanding each exogenous variable's effect on the outcome.

Jung et al. (2018) look at the benefits of using machine learning on traditional data and how this can be a further development of standard econometrics tools for economic forecasting. Their paper addresses the potential of significant errors involved when making predictions and how pre–specified thoughts on relation can lead to biases. Systematic over–prediction or agents' forecasts tend to be "consistently over–optimistic in times of country–specific, regional, and global recessions" (Jung et al. (2018), p. 4) are examples of such errors. Their studies set aside any interpretation of variables and employ the elastic net, Recurrent Neural Network (RNN), and Super Learner. According to their results, Super Learner outperforms the other models with quarterly data, while RNN has some advantage for specific circumstances when forecasting the annual horizon.

The number of studies on machine learning has increased exponentially in the later years. Montgomery et al. (1998) look at the US unemployment rate and touch upon several important features concerning forecasting. They find improvement in using both univariate and multivariate linear models to forecast the US unemployment rate. Moreover, they have a common finding that the mean or median of several models is often more accurate than most individual forecasts – this is in line with the concept of the recent popular model: The Super Learner. The origin of random forest is built on a similar idea according to James et al. saying that "…averaging a set of observations reduces variance and hence increase prediction accuracy. . . ” (James et al. (2013b), p. 316). Varian (2014) finds it strange how rarely the conceptional methods of the Super Learner is exploited in traditional econometrics. It has been recognised for a long time that this method outperforms individual models for macroeconomic model forecasts.

Numerous sources state that inflation is a difficult variable to forecast, one reason being that even central banks have imperfect control over inflation (Svensson (1997)). The article “Forecasting inflation” written by Stock and Watson (1999) studies forecasts of the US inflation over the 1960–1996 period looking primarily at the Phillips curve. Their findings are that going from binary to multivariate models is not necessarily an improvement for inflation forecasting, as adding several variables expected to have good explanatory power on inflation results in overfitting and poor forecast estimates. Stock and Watson’s findings are of interest as we implement machine learning on big data where predictions of inflation are dependent on many variables. We compare these findings with predictions using naïve and AR models where the forecasts are only dependent on past values of inflation.

Makridakis et al. (2018) researched the accuracy of machine learning models compared to traditional statistical methods across multiple forecasting horizons. Their article criticises papers that propose new machine learning algorithms for not comparing the accuracy versus the traditional methods. They are motivated by an article using neural network to forecast stock prices, which got highly accurate results. When Makridakis et al. contacted the authors to get the required information to replicate their results, they were met with radio silence, increasing the suspicion that the results were exaggerated or plain wrong. This indicates that one needs to be critical when trusting papers, and Makridakis et al. concluded that traditional statistical methods are more accurate after reproducing several papers' results. This conclusion is unusual, motivating us to see if machine learning adds value to forecasting purposes.

In response to the article by Makridakis et al., Cerqueira et al. (2019) published a paper criticising their findings. They counter to the results made in the article and question the small size of time–samples used in their research. Cerqueira et al. claim that the traditional statistical methods performing the best are only valid when the sample size is small, otherwise machine learning models outperform. This is an important finding for this thesis as we look at macroeconomic variables published either monthly or quarterly, making our time samples quite short. Therefore, it is interesting to see if the machine learning models manage to perform well even on short time samples.

Another factor in finance and macroeconomics is the low signal–to–noise ratio. With high predictability, the ratio is higher, but the predictability is usually low, especially in finance. Positive news about a company are quicky incorporated into the stock price, thus making the predictability of the low (AQR). Machine learning thrives in high signal–to–noise environments and struggles more in low signal–to–noise environments, making it understandable that machine learning is less trusted in the world of finance. In macroeconomics, the predictability is slightly higher than in finance, as our predicted variables are published on a monthly or quarterly basis, and through using information about events from that month or quartal, machine learning can find conjunctions in the variables.

Artificial neural networks, elastic net, random forest, and Super Learner are recurring machine learning models in almost all relevant literature across different forecast purposes. It is common to emphasise the importance of out–of–sample testing as this is one of the key principles of machine learning models making sure to train the models correctly. This thesis addresses random forest, elastic net, lasso and ridge and a model that averages the result of the mentioned models. We make comparisons of these with the following traditional statistical methods; the naïve, autoregressive, and vector autoregressive models. The studies are applied to a new research area – the Norwegian economy.

3 Methodology and Theory

Machine Learning with large data sets requires knowledge about how to operate and handle data. In classical statistics methods, there often lie assumptions about the underlying distributions of variables. The necessity of complying with these assumptions is not as present in machine learning; however, transforming the data does not harm the model, often it happens to be rather the contrary (Flovik (2018)).

3.1 The Data's Nature

To accommodate the models' underlying assumptions, one must first understand the variables being used in the models. Plotting the variables is helpful to better understand behaviour, patterns and identify outliers in the data set before transforming each variable making sure they are all stationary. We want stationarity because non–stationary variables usually have a clear trend, leading a model to use the previously known variable t as the prediction for $t + 1$. Due to the trend in variables, the model's results might not reflect significant modelling errors, and therefore one might overvalue the precision of the model (Flovik (2018)).

In addition, the different models we use have different assumptions of variables, stationarity being of most importance. To make sure the variables have the desired property of all methods, we test different transformations of the variables. We use the Augmented Dickey–Fuller (ADF), Kwiatkowski–Phillips–Schmidt–Shin (KPSS), and Phillips–Perron (PP) tests to check if the time–series are stationary, trend–stationary and integrated of order 1. Only when the variable has the desired property to satisfy the tests the transformation is accepted. The tests have the following hypothesis,

ADF:

$$H_0 = \text{The variable has a unit root}$$

$$H_1 = \text{The variable is stationary}$$

KPSS:

$$H_0 = \text{The variable is trend-stationary}$$

$$H_1 = \text{The variable has a unit root}$$

PP:

$$H_0 = \text{The variable has a unit root}$$

$$H_1 = \text{The variable is integrated of order 1}$$

One needs to be aware of some pitfalls when handling data. First, looking at macroeconomic time–series data, we must know when the data is published to avoid look–ahead bias and data leak. Given the purpose of forecasting when employing our model, we naturally do not have access to future data. There is a delay of when the quarterly and monthly variables are published. Quarterly data is often published one to two months after the end of the relevant quarter. Therefore, we use only information up to quarter $t - 1$ to enter the model by lagging the time–series data for one period to make sure we only make predictions based on available data.

We want to forecast macroeconomic variables published either monthly or quarterly and use the corresponding data sets when training the model. We convert monthly data to quarterly by using the beginning–of–quarter values. When calculating monthly change for quarterly variables, we take their quarterly difference and divide by three. Some variables have observations as early as 1994, while others only from 2008. When modelling time–series, longer samples of observations often provide more accurate models. We collect 89 variables in our data sets. Given the large number of independent variables, the possibility of the "curse of dimensionality" is present, which can lead to overfitting. This term has roots based on many variables in the input sets often pollutes the models' forecasts rather than help explain the endogenous variable. To avoid this, we look at the trade–off between different combinations of the independent variables in the input data. Another approach to make sure we avoid "curse–of–dimensionality" is to perform Principal Component Analysis (PCA). PCA helps reduce the number of exogenous variables and avoid multicollinearity among the independent variables. A downside of PCA is the loss of interpretation.

Many models do not allow for missing values in the data set. If a variable has missing values, we have a "parted" time–series. To avoid this, we replicated the previous observed value where there is a missing value. By filling in the missing values, we make sure to have time–series with observations in chronological order with equal spread, i.e., one month or quartal between each observation.

Another aspect we need to take into consideration is the bias–variance trade–off. Bias is the difference between the average prediction and the actual value in that period. (Singh (2018)). The variance shows the variability of the estimate when different training data is used. With high bias, the model does not pay much attention to the training set and oversimplifies the model. High variance shows that the model pays too much attention to the training set, leading to poor performance out of sample. The functions below show the relationship between the bias and variance and the total error.

$$Error(x) = (E[\hat{f}(x)] - f(x))^2 + E[(\hat{f}(x) – E[\hat{f}])^2] + \sigma_\varepsilon^2 \tag{1}$$

$$Error(x) = Bias^2 + Variance + IrreducibleError \tag{2}$$

Where $\hat{f}(x)$ is the model used to predict $f(x)$. We disregard the irreducible error since it cannot be avoided. Algorithms such as linear regression tend to have low variance and high bias, while decision trees tend to be high variance, but low bias models (Brownlee (2016)). To find the optimal trade–off between variance and bias errors is important because we do not want a model that overfits or underfits. A model that combines several models could be advantageous in order to lower the bias and variance and find a better trade–off.

3.2 Train and Test Sets

To know whether our model contributes to good forecasts and to avoid overfitting, we need to split the data into three sets. One train set (T_{train}), one test set (T_{test}) and when making predictions we have a third validation set ($T_{validation}$).

$$T \geq T_{train} + T_{test} + T_{validation} \tag{3}$$

The train and test sets contain all the data we need to build our model. When the model is trained, we introduce the validation set to see how well the models work on unseen data. A popular method to train and validate the model is to use a Cross–Validation method that randomly chooses observations for the train and test sets numerous times. However, this method does not work for time–series data as this results in data leak. Also, recent information is often more relevant for future forecasts than information from many years ago. Moreover, we wish to have a time–series sample of previous values in our training and test sets. A solution is to apply a rolling window when deciding on train, test, and validation sets. Starting with a small set, we increase the training sample for each run rolling forward until the entire data set is taken into use, having the test and validations sets of equal and constant size. We introduce the validation set to avoid using out–of–sample to optimize the models. We validate the model by taking the average of precision in predictions of each run. Using this form of Cross–Validation that takes time–series into account helps against the problem of overfitting. When forecasting a quarterly variable one year ahead, we let the test and validation sets consist of four observations ($n = 4$). For monthly data, the test set consists of 12 observations ($n = 12$). To divide the data into train, test, and validation sets, we first split the entire data set into k parts, where $k = N/n$. N is the total number of observations in the data set, and n equals the number of observations in the validation set. In the first run, the train set consists of the first $T_{1:2n}$ observations, the test set consists of $T_{(2n+1):3n}$ and the validation set consists of $T_{(3n+1):4n}$. The train set increases in observations going forward until the last training sample consists of $T_{1:(N-2n)}$, the test set $T_{(N-2n+1):N-n}$ and the validation set $T_{(N-n+1):N}$. This is called an expanding window method and is demonstrated in Figure C.0.1. After finding the models' optimizing measures based on the T_{test} in the models, we merge T_{train} and T_{test} into one train set and train the models again with this sample before introducing the validation set. When forecasting one period we have $n = 1$. We train the model a total of 12 times ($k = 12$), starting with a training set, $T_{1:N-2k}$, test set, T_{N-2k+1}, and a validation set equal to $T_{N-2(k+1)}$, expanding over 12 runs until train equals $T_{1:N-2}$, test equals T_{N-1} and the validation set equals T_N.

3.3 Machine Learning Models

This section presents the different models we implement to research whether machine learning outperforms traditional statistical models in terms of time–series forecasting. First, we employ three different extensions of the ordinary least square regression model (OLS); ridge–, lasso– and elastic net regression. These models are known as penalised regressions, which differ in selecting and shrinking explanatory variables to adjust for complexity in the data. Assumptions for the predictive variable in regression models are linearity, constant variance (no outliers) and independence. The regression models are in the form

$$\hat{y}_t = X_t\hat{\beta} + \varepsilon_t \tag{4}$$

Where $\hat{y}_t$ is a $(n \times 1)$ vector of predictions made for the endogenous variable y for time period t with n observations. X_t is a $(n \times p)$ matrix consisting of p exogenous variables and n observations. $\hat{\beta}$ is a $(p \times 1)$ matrix which are the coefficients made from the regression models, and ε_t is the error term that represents the deviations within the regression line. Next, we test random forest that uses regressions to estimate coefficients in each tree. Finally, we have an Averaging model which takes the average of each machine learning models' forecasts.

3.3.1 Ridge Regression

Ridge regression uses a shrinkage estimator, λ, to shrink the coefficients β towards zero. For $\lambda = 0$, the model equals OLS. The ridge solves

$$\hat{\beta} = \min_{\hat{\beta}} \left(\sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j X_{i,j} \right)^2 + \underset{(1,1)}{\lambda} \sum_{j=1}^{p} \beta_j^2 \right) \tag{5}$$

Where y_i is the endogenous variable at time i, β_0 is the intercept and β_j is the regression coefficient for the exogenous variable j. $x_{i,j}$ is the exogenous variable j's value at time i, these observations are standardised. OLS does not differentiate between the exogenous variables, if multicollinearity and high variance are present it can lead to overfitting. Ridge regression manages to avoid these problems by penalising some of the exogenous variables. The bias–variance trade–off is a motivating factor of ridge regression as a higher λ increases the bias, but at the same time, reduces the variance. Ridge regression is great at handling data set where the number of variables is large compared to the number of observations $(p > n)$. For situations with low signal to noise, ridge can perform well even if the number of observations is large compared to the number of variables (James et al. (2013b) p. 215). When running the ridge and lasso regression models, we apply a sequence of λ varying from 10^{10} to 10^{-10}, over a set of 1,000 values. Usually, the optimal lambda, λ^*, can be found by cross–validation. However, as we are looking at time–series, we choose λ^* to be the one minimizing $RMSE$ based on the test set. After finding λ^*, we combine the train and test set, and train the model with a longer train set using λ^* and then introduce the validation set to compare our predictions.

3.3.2 Lasso Regression

Lasso is very similar to the ridge regression. Instead of using a quadratic penalty, it uses an absolute measure penalty on the form

$$\hat{\beta} = \min_{\hat{\beta}} \left(\sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j X_{i,j} \right)^2 + \underset{(1,1)}{\lambda} \sum_{j=1}^{p} |\beta_j| \right) \tag{6}$$

Description of notation is like the one given in Section 3.3.1. The penalisation term, λ, handles the bias–variance trade–off in the same way it does for ridge, and is computed the same way. Lasso has the property to set some values to zero and therefore eliminate them from the model, which helps the interpretability of the method when resulting in fewer exogenous parameters in the model. (James et al. (2013b) p. 219). Lasso often outperforms ridge when there are many observations ($n > p$).

3.3.3 Elastic Net

Elastic net is a compromise between ridge and lasso regression, with one additional parameter to calibrate, α. Joining the two methods, elastic net has the advantage to take both their strong sides into use. By adjusting α, one can weigh the most appropriate model more than the other. The combination makes elastic net resistant to multicollinearity among the explanatory variables.

$$\hat{\beta} = \min_{\hat{\beta}} \left(\sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j X_{i,j} \right)^2 + \underset{(1,1)}{\lambda} \sum_{j=1}^{p} \left(\alpha \left| \beta_j \right| + (1-\alpha)\, \beta_j^2 \right) \right) \quad (7)$$

This is a similar notation to the one given in Section 3.3.1. When working with elastic net, we are first interested in estimating α to decide whether the model should be weighted more towards ridge or lasso, or if an equal mix ($\alpha = 0.5$) of the two is optimal. We choose the minimum value of λ using the default set in R Studio when searching for the best estimate of α. α is set as a sequence from 1e-06 to 9.999e-01 over a length of 100 points. Next, after finding the optimal weight that minimizes $RMSE$ given the test set, α^*, we run the model once more, now with $\alpha = \alpha^*$ over a set of λ equal to those used on ridge and lasso. Again, we decide λ^* as the one that minimizes the $RMSE$ measure given the test set. These estimates of λ^* and α^* are made in each validation run and used when training the model combining the train and test set before introducing the validation set.

3.3.4 Random Forest

Another approach to forecasting is the classification model random forest. This model has the advantage of no formal assumptions about linearity in the data; hence it can easily be applied without any premodifications. This means that it works well for non–linear data as it is robust to outliers. It runs effectively on large data sets; therefore, we believe random forest works well with our large non–linear macroeconomic data set. The downside is that random forest consists of many steps in its calculations, making it a model with slow computations compared to the other models. However, it does not require many modifications which compensates for its slow computations.

Random forest is a model that origins from the method bootstrap. Bootstrap takes independent draws from the train set and estimates the standard errors of coefficients of a model. The method can be applied to a wide range of statistical estimators as it requires no assumptions. An extension to bootstrap is called bagging, which applies the theory that averaging a set of observations reduces variance. After training a model on the bootstrapped training sets, one takes the average of all the predictions and obtains

$$\hat{y}_{bag}(x) = \frac{1}{P}\sum_{p=1}^{P} \hat{y}^{*p}(x) \tag{8}$$

P is the number of regression trees, and $\hat{y}^{*p}$ are separate prediction models made by the ordinary regression model of the endogenous variable y given x exogenous variables. Hence the model constructs P regression trees using P bootstrapped training sets before taking the average. Finally, random forest is very similar to the idea of bagging; however, each time a split in a tree is considered, it takes a random sample of m out of p predictors to be considered as a candidate before the model uses only one final predictor in the split. By using only one predictor in each split, the models reduce the correlation in each split. Regression trees often have a default of $m = p/3$. If $m = p$ the random forest is identical to bagging (James et al. (2013b) 8.2.2). To decide the optimal number of trees in the model, we apply 100 different tree sizes in each run, from a minimum of one tree to a maximum of 500 trees. We choose the number of trees, $P = M^*$, that minimizes RMSE with respect to the test set. Similar to what we do in the previous models mentioned, we combined the train and test sets running the model on a larger train set on M^* trees with m splits before introducing the validation set.

3.3.5 Averaging Model

In addition to the machine learning techniques, we also create a model that averages the results of the machine learning models (ridge, lasso, elastic net, random forest, and an average of the value y), a technique familiar as “ensemble learning”. It is not as sophisticated as the Super Learner that sets different calculated weights to each model, but averaging models is generally a good idea to accommodate model uncertainty. Weighting several models reduces the chance of overfitting and error variance. The ensemble learning works for every machine learning technique. Therefore, we cannot rule out that different combinations can perform better.

3.4 Statistical Models

To see the value added of machine learning we compare the results with simple traditional statistical models. This thesis makes comparisons with the naïve forecasting model, an autoregressive model, and a vector autoregressive model.

3.4.1 Naïve Model

Naïve forecasting uses the previously observed value as the future forecast.

$$y_t = y_{t-1} \tag{9}$$

The advantage of implementing a naïve model is that it is simple, takes no time to create and requires no assumption of the time–series. However, data in economics are rarely constant; therefore, a significant disadvantage with the naïve model is that no one is likely to trust these predictions unless when looking at highly stable variables. Therefore, this model is mostly added for comparison purposes.

3.4.2 Autoregressive Models

Autoregressive models make predictions using only past values of the endogenous variables and have the assumption that the variables are stationary (Brooks (2014), p. 259). Our models use the Akaike Information Criterion (AIC) to estimate the optimal number of lags, k, which helps deal with the risk of overfitting.

$$AIC = \ln(\hat{\sigma}^2) + \frac{2k}{T} \tag{10}$$

Where $T =$ number of observations, $k =$ number of lags and $\hat{\sigma}^2 = \frac{\sum \hat{u}_t^2}{T-k}$.

3.4.2.1 AR(k) The simplest version of autoregressive models is to use only one endogenous variable, y_t, known as $AR(k)$ which is in the form

$$y_t = \beta_0 + \beta_1 y_{t-1} + ... + \beta_k y_{t-k} + \varepsilon_t \tag{11}$$

It estimates the β coefficients using Yule–Walker equations.

3.4.2.2 Vector Autoregressive Models A VAR model is an extension of the AR model fitting for data sets with more than one dependent variable, where all variables are considered endogenous ((Brooks (2014), p. 335). For g variables, we have a $VAR(k)$ model in the form

$$\underset{(g\times1)}{y_t} = \underset{(g\times1)}{\beta_0} + \underset{(g\times g)(g\times1)}{\beta_1 \; y_{t-1}} + ... + \underset{(g\times g)(g\times1)}{\beta_k \; y_{t-k}} + \underset{(g\times1)}{u_t} \tag{12}$$

With VAR, we allow our predictions of y to depend on more than just its previous values. A downside with this is that it tries to measure everything, and we must therefore make sure that all variables have the same order of integration. OLS and AIC estimate each equation in a VAR to find the optimal number of lags (k). The main issue with VAR is that it requires the number of observations (n) to be larger than the number of variables (p). Therefore, VAR does not work well when training the model over a rolling window as the first training sets consist of few variables. By applying PCA, we try to avoid this problem by reducing the number of exogenous variables. For VAR to work correctly, the solution to the equation above needs to satisfy the PP–test. Cointegration between the variables may exist if y_t has a unit root, meaning that either some or all variables are integrated into order 1.

3.5 Model Comparisons

To evaluate the precision of these models, we use the error measurements Root Mean Square Error ($RMSE$), Mean Absolute Error (MAE), and R^2. $RMSE$ and MAE to evaluate the models' forecasts against the actual data. R^2 reflects the linear relationship, and the closer R^2 is to 1, the better the model's explanatory power. Often in economics, $R^2 = 0.5$ does not qualify as low signal–to–noise, making a lower R^2 measurement more acceptable than "usual" for our models. Impartially, for model comparison, the values themselves are unimportant as we look at their values compared to each other.

$$RMSE = \sqrt{\frac{\sum_{t=1}^{T}(\hat{y}_t - y_t)^2}{T}} \tag{13}$$

$$MAE = \frac{\sum_{t=1}^{T}|\hat{y}_t - y_t|}{T} \tag{14}$$

$$R^2 = \frac{SSM}{SST} = 1 - \frac{SSE}{SSM} \tag{15}$$

$RMSE$ represents the standard deviation of the residuals. In contrast, MAE represents the absolute average size of the residuals, and the R^2 represents the model sum of squares (SSM) divided by the Total Sum of Squares (SST). The values of $RMSE$ and MAE are scaled and the same unit as the dependent variable that we forecast. $RMSE$ punishes large deviations from actual value more by taking the quadratic error which MAE does not. Due to this fact and that $RMSE$ lacks some direct interpretation, it is helpful to look at both measures. In cases where two models have similar MAE but different $RMSE$, one model will have more significant deviations, although the errors are similar on average. In forecasting macroeconomic variables, precision is essential, and we wish to avoid large deviations, making $RMSE$ a good measure. Kathuria (2019) argues that the Mean Square Error (MSE) is the best error measure and that $RMSE$ is an even better measure as it takes the root, which makes the measure similar in unit and size to our predictive variable. Especially for error measures, it is critical to have fixed the problem of non–stationarity to avoid misleading and incorrect measurements (Flovik (2018)). Also, we look at the average of error measurements in each validation set.

Final validation of the models' performances is done by testing for robustness. This can be done by testing the models on different periods, especially with and without a crisis in the data set. We look at predictions including and excluding Covid–19 and test the models' long– and short–term predictions. Since we look at different data sets, we can decide whether a consistent model outperforms the other; however, we do not compare model performance across different data sets.

4 Data

The data must cover all aspects of the economy to ensure that our model can observe and register information of some explanatory powers. The data must also be sufficiently long to include the different economic cycles and the different variables. The longer the sample of observations in the data set – the greater possibilities for a better model. It is a common problem in economics that many variables lack series that are sufficiently long, and we face the issue of accessibility of long data samples. An alternative is to remove variables with too short time–series from the data set. However, we risk losing variables with explanatory power of our dependent variable—this thesis explores where the ideal trade–off between length and number of variables lies.

Through help from Nordea Markets' Macroeconomics team, we collect 89 variables through the database Macrobond, a database that helps collect, analyse, and visualise data. The database has all the variables we find relevant for our thesis, and therefore we only used this tool as our data collector. The variables are collected in either monthly or quarterly frequency. These data include our three endogenous variables, GDP, unemployment rate and inflation, and other variables such as import and export from different countries, Indexes, FX rates, Swap rates, Government and Corporate Bond Rates, and Economic Surveys. All variables are disclosed in F.1.1 with their transformations, descriptions, and frequency. We take the logarithm of all variables with values > 0 and, for all variables, difference them until they reject (for KPSS – not reject) the unit–root tests at a 99% confidence level. We lag monthly and quarterly variables to avoid look–ahead bias. It is not necessarily the sample length of each series that is most valued. Having many features with little correlation is often sufficient to find the true pattern in the data when applying machine learning.

After handling the raw data as disclosed in Section 4 we look at different combinations of data sets. When removing missing values from the entire data set, we are left with observations from 2008 to 2021. Removing the three variables with the most missing values, we have a data set that goes back to 2005, and by setting a threshold of missing values less than 25 for the quarterly data set and less than 50 for the monthly data set, we have data back to 2000 and 1998 respectively. Even though removing several variables helps avoiding the “curse–of–dimensionality”, we might remove a vital exogen variable. On the other hand, there might be several variables with little explanatory power; hence by performing PCA on all data sets, we can test whether additional columns are removed or not. We run the PCA using three different tolerance thresholds of a minimum level of variance: 0.1, 0.05 and 0.01. Any subsequent component with standard deviation, σ, less than $tol \times \sigma_{PC1}$ are not included in the analysis of principal components. The combination of removing variables with large numbers of missing values and performing PCA, we are left with nine data sets to test our models.

5 Results and Analysis

5.1 Results

To validate the assumptions for the penalised regression models, we check that the data is linear, normally distributed, and independent. Figure I.0.1 shows the distribution of the endogenous variables before and after being standardised, and we see a clear improvement in behaviour of the variables after. This is valid for the exogenous variables as well, being linear or almost linear, which is appropriate for our models as they are extended versions of OLS. Even though linear regressions assume linearity, our models use a penalisation term to operate around this and therefore work well with non–linear data. The variables have constant variance as we have made sure the variables are stationary and trend–stationary at a 99% confidence interval. Figure G.0.1 shows all the endogenous variables' autocorrelation function plots first on their raw form to the left and after their transformations to the right. We see a clear improvement in the time–series autocorrelation among the lagged variables after the transformation, indicating that their observations are now independent. This property is checked for all variables. Moreover, due to the PCA, we are confident that all the principal components used in the models are independent as it removes multicollinearity. Either way, we also see little correlation in the data sets by Figure H.1.1 and Figure H.2.1, meaning most variables are independent in their raw forms.

Figure D.0.1 shows the regression coefficients in dependence of λ. We see that several coefficients are set to zero in the lasso regression model (to the right) for an increasing λ, while the ridge coefficients (to the left) assume small but non–zero values for large λ, both observations are characteristics for general solutions of the two regression models. For the lasso regression model, we have that the coefficients for $log(\lambda) > 0$ are penalised, in this plot the optimal λ, λ^*, (minimizing $RMSE$) equals 0.0032 ($log(0.0032) = -2.4949$) and one out of 37 coefficients is penalised. In the ridge regression λ^* is a bit larger and equals 0.0177, $(log(0.0177) = -1.7520)$.

To verify that the number of candidates at each split, m, in the random forest model is optimal when $m = p/3$ we tested for $m = p$, $m = p/3$ and $m = \sqrt{p}$. Figure E.0.1 plots of the three different models with the corresponding different number of candidates over a grid of many trees and the corresponding $RMSE$. We verify that $m = p/3$ is reaching a value below the two others as the number of trees increases. From Figure E.0.1, we also see that the $RMSE$ decreases as the number of trees increases.

The predictions made of the naïve model are only the last known value of y in the last training set replicated n times where n equals the length of the validation set. The predictions made from the naïve models are constant, thus no variance, and we can therefore not measure R^2. However, there are some similar occurrences in some of the training sets for the other regression models. For these incidences, we sat $R^2 = 0$, which gives the models an overall lower estimate of R^2 when taking the average of the runs.

For the autoregressive model we make sure not only the endogenous variables, but all variables are stationary and trend–stationary for model validation purpose. Before running the AR model, we find an optimal number of lags using AIC, resulting in optimal number of lags equal, $k = 1$. Therefore, we have $AR(1)$ model in the form

$$y_t = \beta_0 + \beta_1 y_{t-1} + \varepsilon_t \tag{16}$$

The VAR model is not functional on data sets with fewer observations than variables, resulting in no computations of VAR in these cases. To avoid this problem, we first tried to increase the number of endogenous variables as the number of observations increased in the training sample, letting the number of principal components used in VAR equal $p = x - 1$ for x observations.

Unfortunately, this led to large error measurements. Instead, we skipped the VAR model until the length of the training set $T_{1:x}$ is longer than the number of variables ($x > p$). Therefore, VAR has been trained fewer times than the others, and as a result, its error measurements are likely to be relatively large.

To analyse the models, we look at each model's average one–year predictions from 2010 to 2019. In addition, we wish to see how the models perform in the short term, therefore we also included the results on the models' performance on a one–period prediction taking the average of 12 runs, including and excluding Covid–19. To check robustness, we look at their one–year performance of 2019 and 2020 to validate whether the models' results before and after Covid–19 are consistent with the average results. It is worth noticing that the error estimates of the quarterly variables (GDP and unemployment rate) in the annual predictions are based on four predictions points. As inflation is a monthly variable, the error estimates are based on 12 prediction points; hence, more extensive error measurements when forecasting inflation are expected.

5.1.1 Norwegian GDP, Quarterly Predictions

The best combination of data is to apply all variables in our data set, creating a trade–off with a shorter time sample with data from 2008 and tolerance equal to 0.1 in the PCA, meaning that any principal components with standard deviation less than $0.1 \times \sigma_{PC1}$ are removed from the analysis. This results in our model running with 37 principal components on 52 quarterly observations ($n > p$).

The error estimates of the predictions are listed in Table A.1.1. There is some variation in what models perform the best when forecasting different periods based on different data sets. Running the models ten times, forecasting one year excluding Covid–19, the lasso model has on average the lowest $RMSE$ and MAE estimates, closely followed by the Average ML model and elastic net. The ridge regression has the highest R^2 measurement, while the lasso has relatively low R^2 given that it outperforms on the other two measures. $AR(1)$ performs the best of the traditional statistical methods and even outperforms random forest for all three measurements.

The error measurements for one year of forecasting in 2019 the naïve model has the lowest $RMSE$, closely followed by the penalised regressions. $AR(1)$ has the lowest MAE measurement. Again, the penalised regression models' MAE are close. The machine learning models have high R^2 (except lasso), with elastic net having the highest score. Forecasting the year of Covid–19, the penalised regressions showed some muscle having the lowest $RMSE$ and MAE and the highest R^2. In general, the forecasting results of this turbulent year are comparatively similar to the average one–year forecasts.

On the one–period forecast excluding Covid–19, the $AR(1)$ outperform the other models, followed by random forest. Forecasting one quartal including Covid–19, the Average ML model perform the best.

5.1.2 Norwegian Unemployment Rate, Quarterly Predictions

The best combination of data is the data set including all variables and with tolerance equal to 0.01 in the PCA, thus components with standard deviation less than $0.01 \times \sigma_{PC1}$ are removed from the analysis, leaving the data set to consist of 51 principal components and 52 quarterly observations (including date and NOUR, $n < p$). Since we have fewer observations than variables, $VAR(1)$ could not run.

The results for the Norwegian unemployment rate forecasting error measurement are given in Table A.2.1. When excluding Covid–19 from the data set, $AR(1)$ is on average the model with closest predictions compared to the true data having lowest $RMSE$ $(= 0.8876)$ and MAE $(= 0.7592)$. We have seen that the machine learning models (excluding ridge) have the highest R^2 estimates. Average ML model is the second–best model, with $RMSE = 0.9254$ and $MAE = 0.7828$. Random forest has the highest $RMSE$ and MAE measurements and the highest R^2 of all models.

lasso outperforms all other models on all metrics when forecasting only 2019. Overall, the other machine learning models do quite well, outperforming naïve and $AR(1)$. When forecasting the year of Covid–19, the models' performance is more divided. The $AR(1)$ has the lowest $RMSE$ (= 2.0660), ridge has lowest MAE (= 1.8313) and random forest has the highest R^2 (= 0.9032). Ridge has the second–lowest $RMSE$ (= 2.1238), however a very low R^2 (= 0.0988), while $AR(1)$ has the second–lowest MAE (= 1.8943) and a higher R^2 (= 0.6499).

On the one–period forecasts excluding Covid–19, $AR(1)$ is most accurate, followed by Average ML. The forecasts over the year of Covid–19, ridge outperforms the others followed by elastic net. Random forest has the highest MAE when forecasting unemployment rate on short term.

5.1.3 Norwegian Inflation, Monthly Predictions

Forecasting inflation the data set with all variables included is the best combination, with observations from 2008. The optimal tolerance in the PCA is 0.01, resulting in zero components being removed. Thus, the data set consists of 87 principal components and 159 monthly observations ($n > p$).

The inflation prediction results are shown in Table A.3.1. $AR(1)$ outperforms the other models in every single prediction period, with all three error measures giving consistent results. The second–best model varies among the machine learning models, where lasso and random forest are often closest to $AR(1)$. Average ML is the second–best when forecasting one year on average.

5.2 Analysis

To analyse the results, we look at consistency in the models. We are interested in whether the same models do well across the endogenous variables, time horizons and during recessions.

As inflation is a monthly variable, the one–year forecast is based on 12 prediction–points. The greater the prediction steps, the greater uncertainty, so we are expecting large error measures. $AR(1)$ clearly outperforms the others when forecasting inflation, as the results are very consistent. Given that $AR(1)$ and naïve depend only on inflation in their predictions, while machine learning uses other macroeconomic factors as exogenous variables, our results are in agreement with the findings of Stock and Watson (1999). In addition, the standard deviation of inflation reported in Table B.0.1 are very stable and quite low. We can not show that machine learning adds value when forecasting this variable and believe that using many variables to explain inflation is more confusing than value–adding.

On average, the results for GDP show that the machine learning models with lasso and the penalised regressions perform the best. The best statistical method, $AR(1)$, perform 12.06% worse according to the $RMSE$ than the best machine learning model, and 10.8% worse according to MAE. For the unemployment rate, the $AR(1)$ model perform the best, however, in relation to the penalised regressions it is only 4.25% better according to RMSE and 3.1% better according to MAE. It is clear that $AR(1)$ only slightly outperforms the other models.

Forecasting a specific period helps test for robustness in the models. On the one–year forecast of 2019, there are some variations in the models where the traditional statistical models have lower error measures for GDP and lasso the lowest for unemployment rate, not truly consistent with the one–year results on average. In Table B.0.1 we see that GDP is very stable during 2019 with a standard deviation equal to 0.51%. From Figure A.1.1 we see that the period before 2019 is also very stable compared to the standard deviation of 2010–2020 (0.90%). Therefore, it makes sense that the traditional models are doing well for this period. We suddenly have opposite results for the unemployment rate, machine learning now clearly outperforming the traditional models. Figure A.2.1 shows a quite stable period of unemployment rate right before 2019 and a relatively volatile period during 2019. We see from Table B.0.1 that this is a more volatile period having a standard deviation equal to 14.41% compared to the average of 9.46%, this could be a natural reason why machine learning performs better.

Introducing a volatile period such as Covid–19 is of interest to validate what models manage to maintain their level of performance during recessions. When looking at GDP, the results are consistent with the one–year forecasts on average, lasso having the lowest $RMSE$ and MAE, and the machine learning models have high R^2. The model performances for unemployment rate are also very similar, except that ridge has now the lowest measure of MAE. Looking at the periods' volatility in Table B.0.1 we see high volatility in 2020 for both variables. In 2020 the unemployment rate an extremely high standard deviation of 29.60%, and GDP of 6.19%. Therefore, it makes sense that machine learning continues to do well for GDP and that there is an improvement when looking at MAE for the unemployment rate.

Looking at the models' performance for one–period forecast excluding Covid–19, $AR(1)$ perform the best for both unemployment rate and GDP, Average ML and random forest did respectively perform second best. The error measure differences are 22.6% and 14.24%. A noticeable result is that the elastic net and lasso are outperform by the naïve model when forecasting GDP. The machine learning models are more precise when including Covid–19. Average ML and ridge perform best for GDP and unemployment rate, respectively. The $AR(1)$

model perform 5th and 4th, with 32.7% and 56.6% worse error measures than the best machine learning model. The machine learning models better manage to follow the oscillation of the Covid–19 period 2020, and therefore, the results changed drastically.

Interestingly, when $AR(1)$ outperforms the machine learning models, the difference in error measures is relatively low. The machine learning models do not perform poorly even though the $AR(1)$ performs the best. Further, when the machine learning models outperform the statistical methods, the differences in error measures are relatively higher. In conclusion, we find the machine learning models to have more sound forecasts in general.

Random Forest, elastic net and ridge are good at handling data sets with many variables compared to observations. Random forest should be able to sort a wide spectre of exogenous variables to extract non–linear relationships. Elastic Net through combining the dimension reduction and variable selection and produce results that are robust to handle multicollinearity Tiffin (2016). The fact that random forest performs poorly, and the penalised regressions well for one–year forecasting could be due to the somewhat linear relationship between the endogenous and exogenous variables.

The Average ML model's performance naturally depends on the input models. Significant outliers in models are not ignored but included with the same weight as models that perform superior. In this case, the random forest model performs relatively poorly, negatively affecting the Average model's result. More advanced algorithms decide the weighting of the input models in a more complex way than taking the average of predictions. Thus, the results could be even better by including an advanced model, such as the Super Learner.

Comparing our results with the ones of Makridakis et al. (2018) and Cerqueira et al. (2019) is quite interesting. As Cerqueira et al. stated, having a short time–sample, the statistical methods are expected to perform better than machine learning. We consider our longest data set with monthly observations of inflation to be quite short, and the results from this data set are that the statistical models outperform the other. Our result is consistent with the findings of Cerqueira et al. however, as previously addressed, we believe this to be due to the behaviour of inflation and not a validation for statistical models, in general, doing well with short time samples. For the shorter quarterly data sets, there is no consensus on what model is performing the best, and often machine learning models are the ones with the elite forecasts. This result is neither in agreement with the paper from Cerqueira et al. nor Makridakis et al. With a short–time sample, we should, according to Cerqueira et al, get the same results as Makridakis et al., which is not the case.

6 Conclusion

This thesis provides new measures of comparing machine learning to traditional statistical methods in time–series forecasting of Norwegian Macroeconomic variables over one–year and one–period horizons. The results are based on standardised, stationary macroeconomic and financial data. To evaluate the models out–of–sample avoids overfitting, and we find the optimal trade–off between exogenous variables and observations by performing PCA. The primary conclusion of this study is that machine learning does add value when forecasting macroeconomic variables, excluding inflation. Similar to Stock and Watson (1999), our study finds inflation to be a complex variable to predict, and the models depending solely on inflation itself are consistently outperforming the others. Machine learning models perform best when predicting GDP and their performances are validated when testing for robustness. When forecasting the unemployment rate, the outperforming results vary between the machine learning and statistical models. However, the traditional methods are never better to a great extent. Again, the results are similar to when forecasting the unemployment rate during the period of Covid–19, hence validating the robustness of the models. In situations where machine learning outperforms, there is truly a significant improvement compared to traditional statistical methods. In situations where the statistical methods outperform, the difference is limited compared to the results from the machine learning models. This gives us reason to conclude that using machine learning instead of traditional statistical methods adds value for forecasting purposes. When looking at volatile periods we find machine learning to have a great advantage as it captures abnormalities and rapid fluctuations.

Further research can create more complex traditional statistical methods and use even more advanced machine learning models, such as Super Learner or dive more into the Deep Learning sphere and use Recurrent Neural Network.

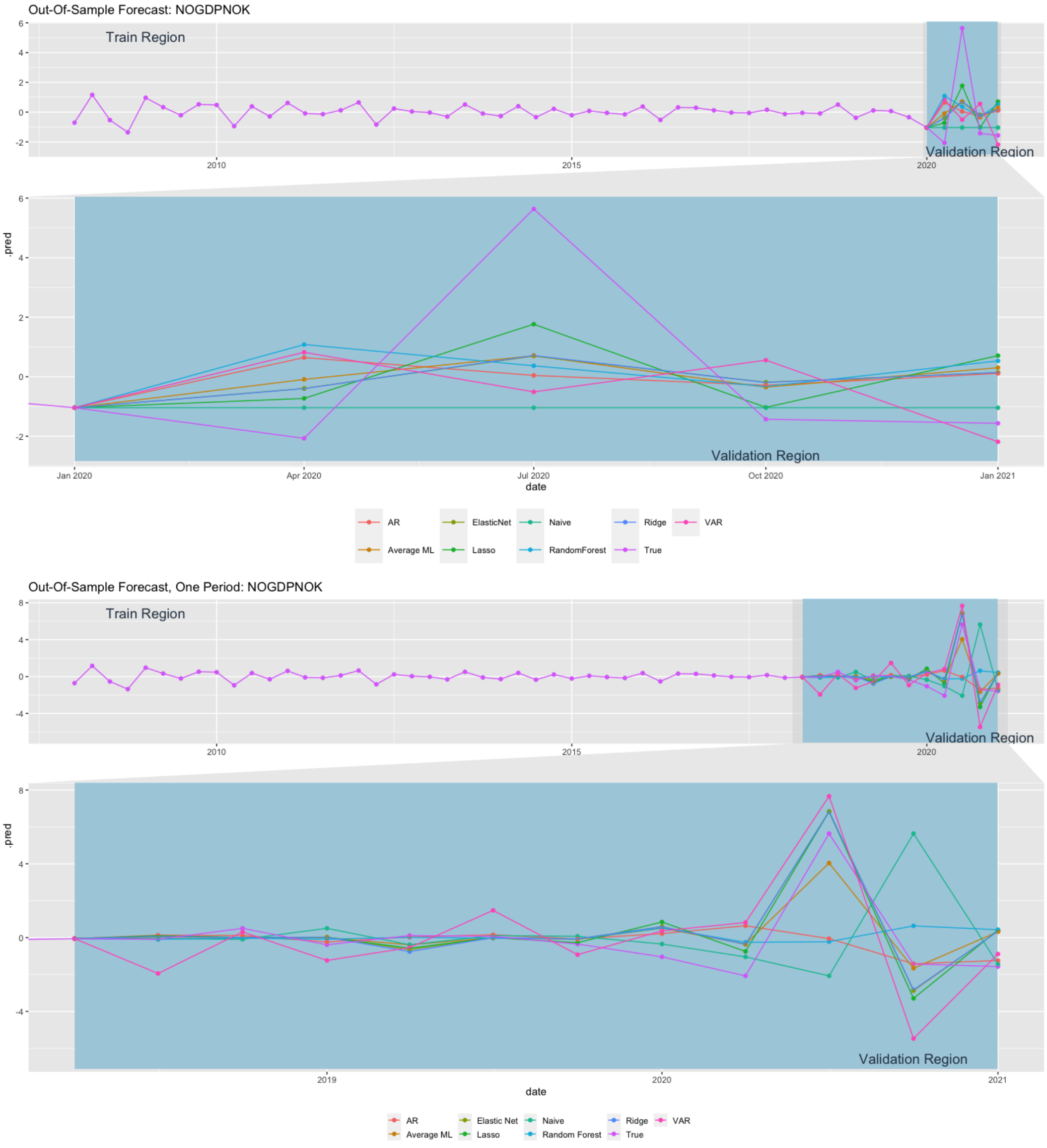
Also, testing the models on other macroeconomic or financial factors could be interesting. Unfortunately, we cannot state that the result will not change if using other methods, adding, or excluding variables or tuning the models differently.

Appendices

Appendix A Model Validation

A.1 Norwegian Gross Domestic Product Mainland

Figure A.1.1: Out of Sample Forecast: Norwegian GDP



Note. The top plot is out of sample forecast of Norwegian GDP (NOGDPNOK) one year ahead (four quarters) and the lower plot is one period forecast (one quarter) for 12 periods plotted together. The purple line is the true values and the other are the different models' forecasts.

Table A.1.1: Model Performance Metrics, Norway GDP

NORWEGIAN GROSS DOMESTIC PRODUCT MAINLAND

ONE YEAR FORECAST ON AVERAGE: Jan 2010–Jan 2020

	Ridge	Lasso	Elastic Net	Random Forest	Average ML	Naïve	AR(1)	VAR(1)
$RMSE$	0.7023	0.6376	0.6576	0.7632	0.6560	0.7955	0.7145	0.8925
R^2	0.4659	0.2733	0.2691	0.1986	0.2912		0.3392	0.1605
MAE	0.5736	0.5284	0.5462	0.62	0.5330	0.7046	0.5856	0.8321

ONE YEAR FORECAST: Jan 2019–Jan 2020

	Ridge	Lasso	Elastic Net	Random Forest	Average ML	Naïve	AR(1)	VAR(1)
$RMSE$	1.1492	1.0723	1.1492	1.1993	1.137	0.967	1.1149	1.1703
R^2	0.5103	0.0537	0	0.4595	0.2504	.	0.1617	0.0894
MAE	0.8054	0.7612	0.8054	0.8414	0.8038	0.8477	0.7597	1.0683

ONE YEAR FORECAST COVID–19: Jan 2020–Jan 2021

	Ridge	Lasso	Elastic Net	Random Forest	Average ML	Naïve	AR(1)	VAR(1)
$RMSE$	2.8066	2.352	2.8154	3.291	2.8681	3.3947	3.2699	3.5523
R^2	0.8238	0.6676	0.8205	0.0143	0.6522		0.0466	0.0136
MAE	2.3856	1.9718	2.3929	2.905	2.4651	2.1551	2.7813	2.9108

ONE PERIOD FORECAST: Jan 2017–Jan 2020

	Ridge	Lasso	Elastic Net	Random Forest	Average ML	Naïve	AR(1)	VAR(1)
MAE	0.4967	1.0386	1.1214	0.3813	0.6167	0.6336	0.3109	1.0023

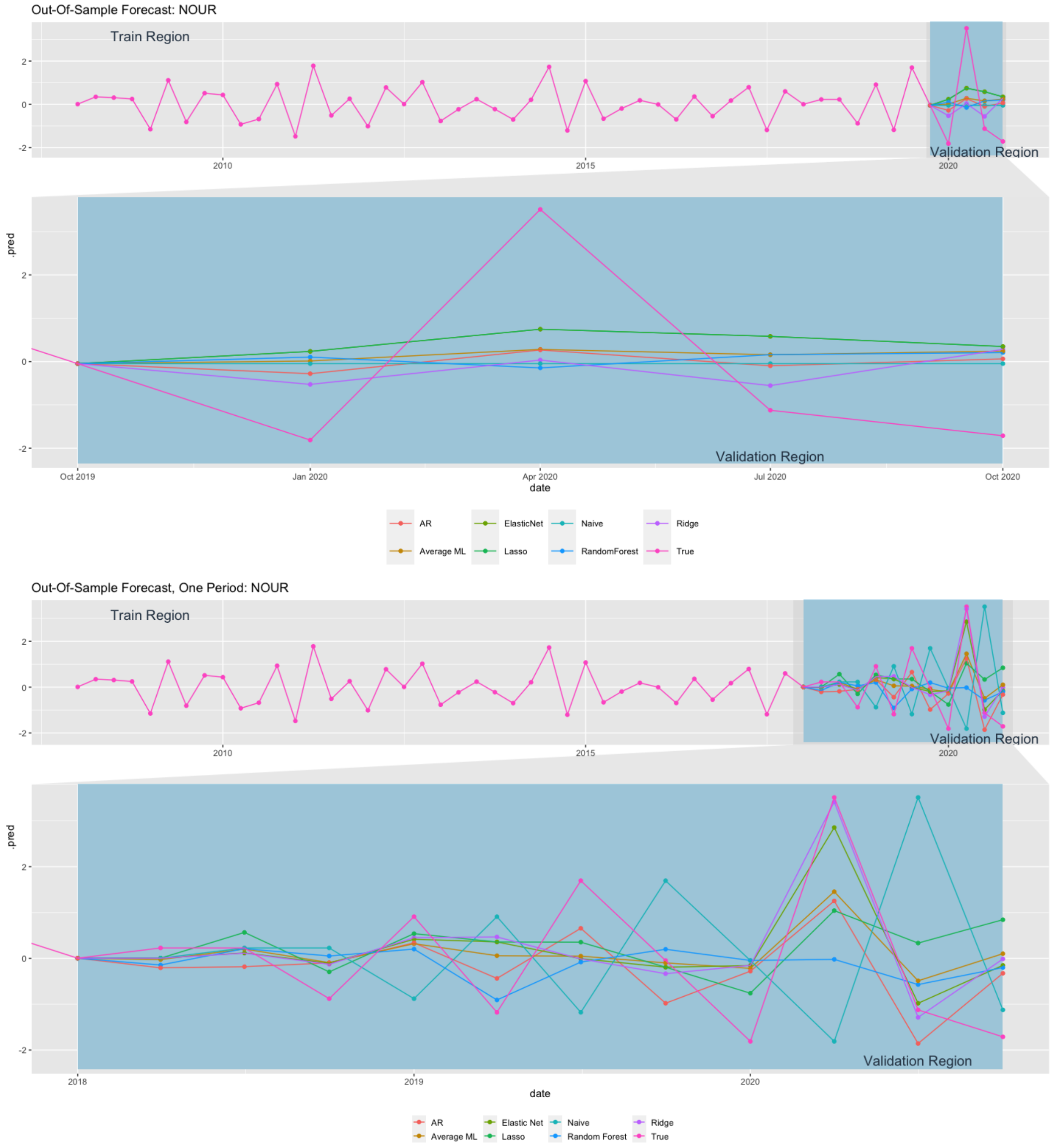
ONE PERIOD FORECAST COVID-19: Jan 2018–Jan 2021

	Ridge	Lasso	Elastic Net	Random Forest	Average ML	Naïve	AR(1)	VAR(1)
MAE	0.8216	0.8202	0.8143	1.262	0.6948	1.8987	1.0882	1.4441

Note. This table reports the model performance metrics $RMSE$, R^2 and MAE we have applied to evaluate the different models. The *One year ahead forecast: Jan 2020–Jan 2021* data set's metrics corresponds to the top plot in A.1.1, while *One period forecast: Jan 2018–Jan 2021* to the lower plot. We also tested the model performances on average with full data sample without Covid–19 and the one year forecast of 2020.

A.2 Norwegian Unemployment Rate

Figure A.2.1: Out of Sample Forecast: Norwegian Unemployment Rate



Out-Of-Sample Forecast: NOUR

Out-Of-Sample Forecast, One Period: NOUR

Note. The top plot is out of sample forecast of Norwegian Unemployment Rate (NOUR) one year ahead (four quarters) and the lower plot is one period forecast (one quarter) for 12 periods plotted together. The purple line is the true values and the other are the different models' forecasts. Note that VAR does not work on this data set, hence true data is now demonstrated by the pink line.

Table A.2.1: Model Performance Metrics. Norway Unemployment Rate

NORWEGIAN UNEMPLOYMENT RATE

ONE YEAR FORECAST ON AVERAGE: Jan 2010–Jan 2020

	Ridge	Lasso	Elastic Net	Random Forest	Average ML	Naïve	AR(1)
$RMSE$	0.9491	0.9273	0.9482	0.9642	0.9254	1.3723	0.8876
R^2	0.2320	0.2086	0.1942	0.5453	0.3421		0.5177
MAE	0.8037	0.7910	0.8127	0.8186	0.7828	1.2299	0.7592

ONE YEAR FORECAST: Jan 2019–Jan 2020

	Ridge	Lasso	Elastic Net	Random Forest	Average ML	Naïve	AR(1)
$RMSE$	1.2825	1.0378	1.2766	1.6549	1.3509	2.1522	1.4645
R^2	0.5163	0.7038	0.5217	0.0182	0.5625		0.3637
MAE	1.1065	0.9403	1.1094	1.5790	1.1910	1.9333	1.2314

ONE YEAR FORECAST COVID–19: Oct 2019–Oct 2020

	Ridge	Lasso	Elastic Net	Random Forest	Average ML	Naïve	AR(1)
$RMSE$	2.1238	2.1834	2.1836	2.3655	2.1946	2.2199	2.0660
R^2	0.0988	0.7289	0.7294	0.9032	0.4011		0.6499
MAE	1.8313	2.1491	2.1493	2.1941	2.0751	2.0162	1.8943

ONE PERIOD FORECAST: Jan 2017–Jan 2020

	Ridge	Lasso	Elastic Net	Random Forest	Average ML	Naïve	AR(1)
MAE	0.8182	0.8309	0.8514	0.9647	0.7998	1.6731	0.7001

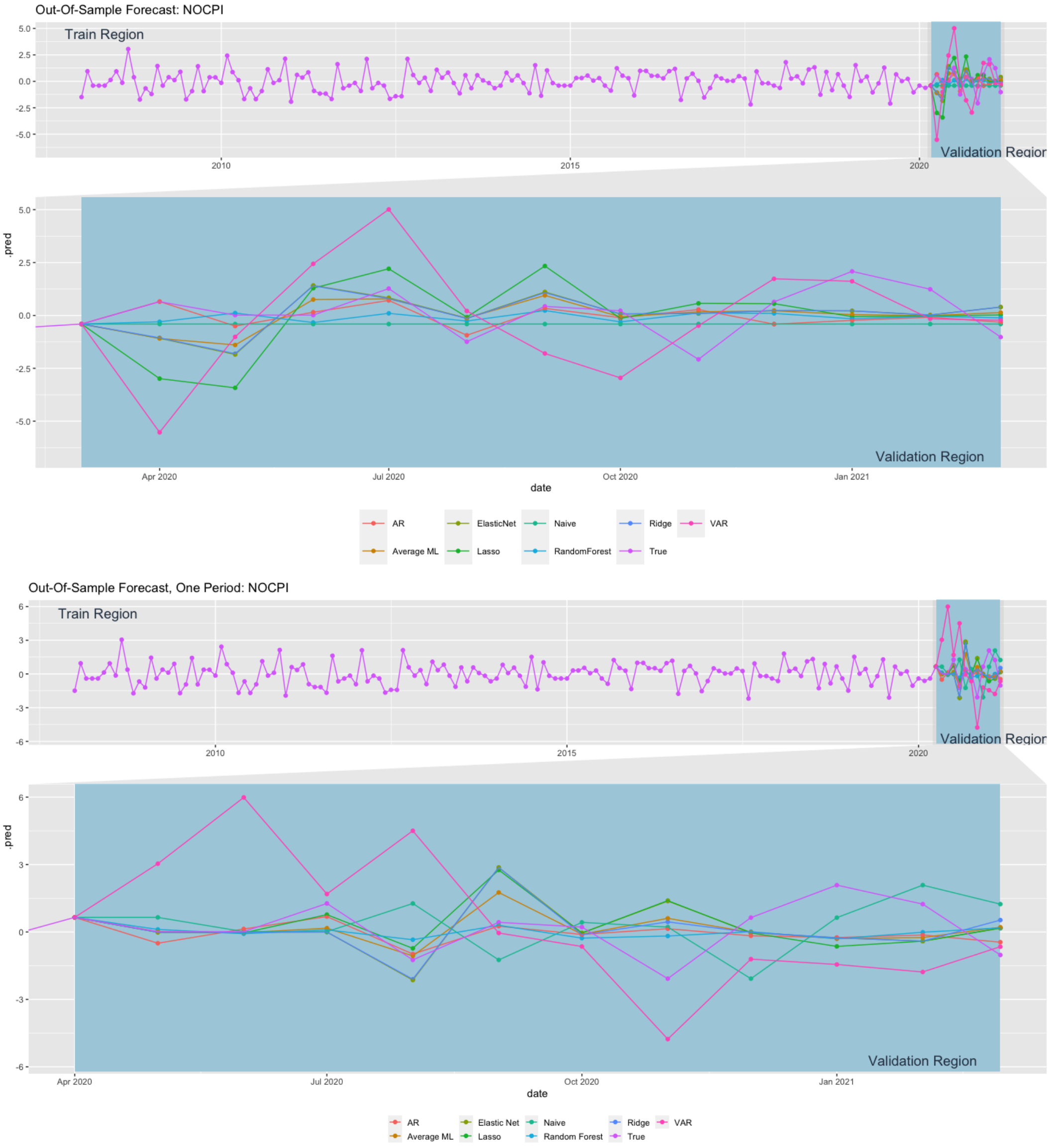
ONE PERIOD FORECAST COVID–19: Jan 2018–Oct 2020

	Ridge	Lasso	Elastic Net	Random Forest	Average ML	Naïve	AR(1)
MAE	0.7106	0.9520	0.7436	1.0171	0.8871	2.1548	0.9431

Note. This table reports the model performance metrics $RMSE$, R^2 and MAE we have applied to evaluate the different models. The *One year ahead forecast: Jan 2020–Jan 2021* data set's metrics corresponds to the top plot in A.2.1, while *One period forecast: Jan 2018–Oct 2020* to the below plot. We also tested the model performances on average with full data sample and without the time period of Covid–19.

A.3 Norwegian Inflation

Figure A.3.1: Out of Sample Forecast: Norwegian Inflation



Note. The top plot is out of sample forecast of Norwegian Consumer Price Index (NOCPI) one year ahead (12 months) and the lower plot is one period forecast (one month) for 12 periods plotted together. The purple line is the true values and the other are the different models' forecasts.

Table A.3.1: Model Performance Metrics, Norway Inflation

METRICS NORWEGIAN INFLATION (CPI)

ONE YEAR FORECAST ON AVERAGE: Mar 2010–Jan 2020

	Ridge	Lasso	Elastic Net	Random Forest	Average ML	Naïve	AR(1)	VAR(1)
$RMSE$	0.8408	0.8642	0.8309	0.8973	0.8053	1.4416	0.7856	1.8662
R^2	0.2439	0.2138	0.2415	0.1160	0.2596		0.3021	0.1234
MAE	0.6857	0.7098	0.6905	0.7273	0.6695	1.2702	0.6448	1.5492

ONE YEAR FORECAST: Jan 2019–Jan 2020

	Ridge	Lasso	Elastic Net	Random Forest	Average ML	Naïve	AR(1)	VAR(1)
$RMSE$	0.8134	0.7839	0.8128	1.3937	0.8656	1.7452	0.6329	1.3272
R^2	0.3294	0.3833	0.3301	0.0008	0.2467		0.6294	0.1453
MAE	0.6784	0.6536	0.6774	1.0738	0.7297	1.5459	0.5300	0.0799

ONE YEAR FORECAST COVID-19: Mar 2020–Mar 2021

	Ridge	Lasso	Elastic Net	Random Forest	Average ML	Naïve	AR(1)	VAR(1)
$RMSE$	1.3577	1.8728	1.3537	1.1643	1.2681	1.2664	1.1209	2.6273
R^2	0.0046	0.0016	0.0048	0	0.0031		0.0644	0.0385
MAE	1.2021	1.5813	1.2004	0.9527	1.1067	1.1118	0.8083	2.1319

ONE PERIOD FORECAST: Jan 2019 - Jan 2020

	Ridge	Lasso	Elastic Net	Random Forest	Average ML	Naïve	AR(1)	VAR(1)
MAE	0.8530	0.7663	0.8565	1.044	0.8265	1.4131	0.5181	1.2535

ONE PERIOD FORECAST COVID-19: Mar 2020 - Mar 2021

	Ridge	Lasso	Elastic Net	Random Forest	Average ML	Naïve	AR(1)	VAR(1)
MAE	1.2134	1.2208	1.3021	0.9044	1.0171	1.3588	0.8709	2.5079

Note. This table reports the model performance metrics $RMSE$, R^2 and MAE we have applied to evaluate the different models. The *One year ahead forecast: Mar 2020–Mar 2021* data set's metrics corresponds to the top plot in A.2.1, while *One period forecast: Mar 2020–Mar 2021* to the below plot. We also tested the model performances on average with full data sample and without the time period of Covid–19.

Appendix B Volatility

Table B.0.1: Volatility of GDP, Unemployment Rate and Inflation

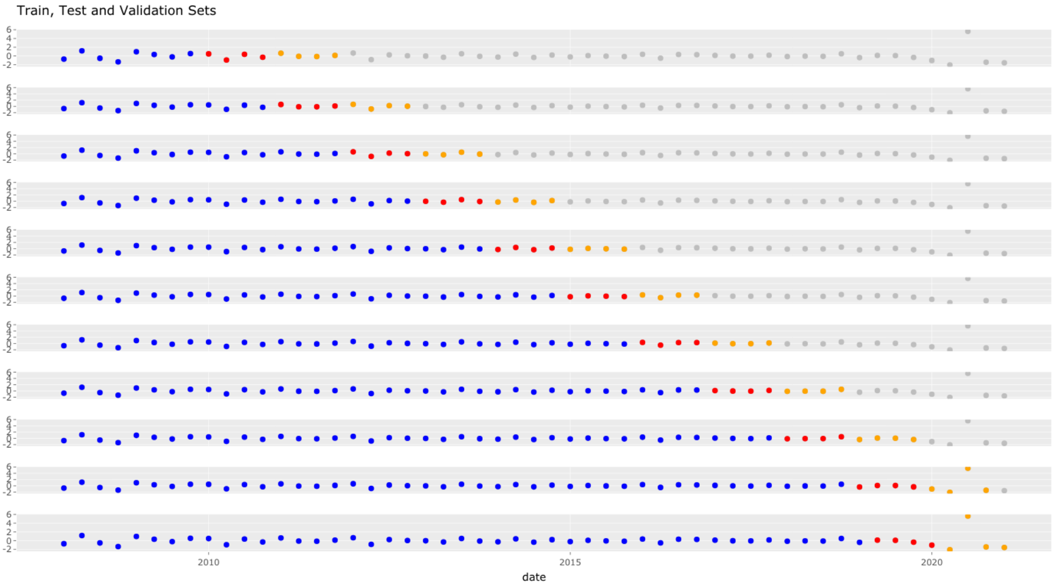
STANDARD DEVIATION

	GDP	**UR**	**CPI**
Average	0.90 %	9.46 %	0.42 %
2019	0.51 %	14.41 %	0.47 %
2020	6.19 %	29.60 %	0.44 %

Note. The standard deviation of the endogenous variables on over the long sample with average predictions and for the robustness test periods 2019 and 2020.

Appendix C Train, Test and Validation Sets

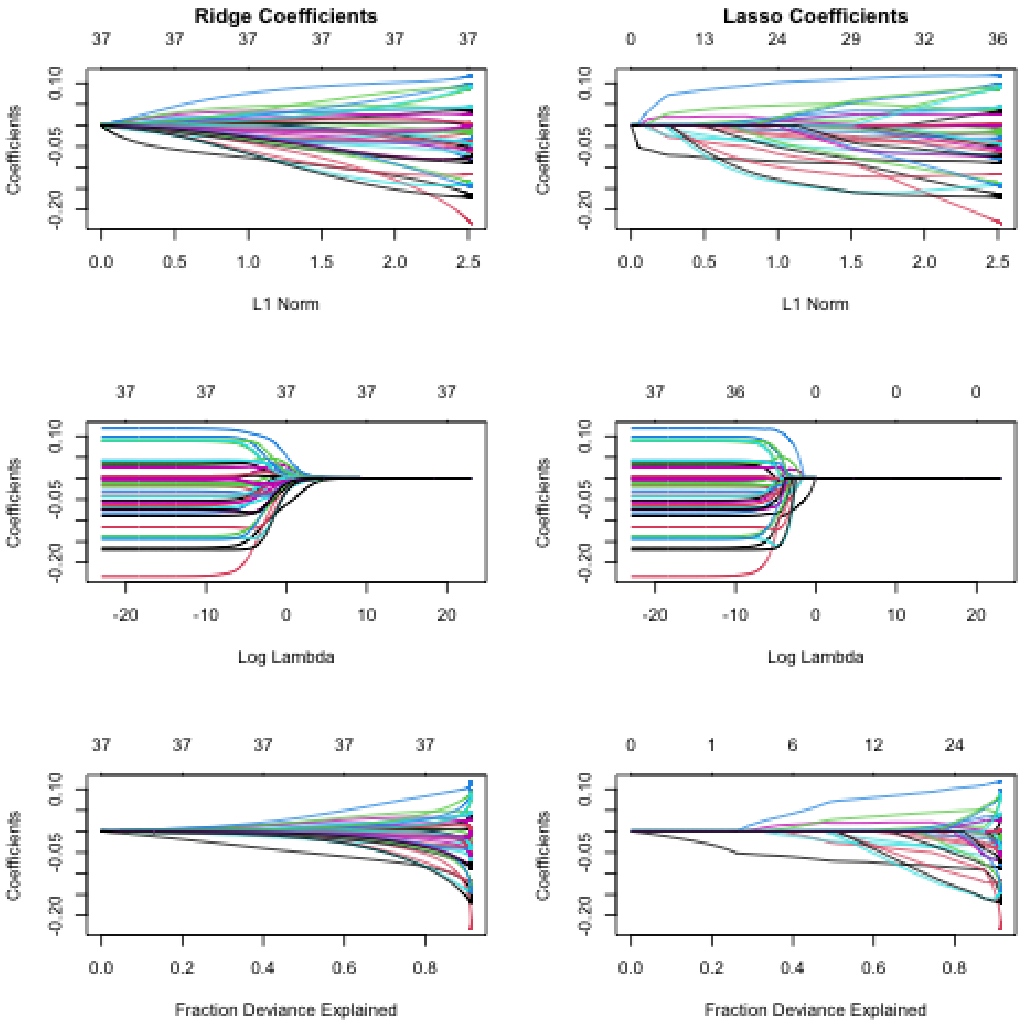
Figure C.0.1: Train, Test and Validation Sets



Note. An illustration of the train, test and validation sets over an expanding window. The blue part is the train set, the red is the test set and the orange points are the validation set. We see that the train set expands while the test and validation sets are of constant size.

Appendix D Ridge and Lasso Coefficients

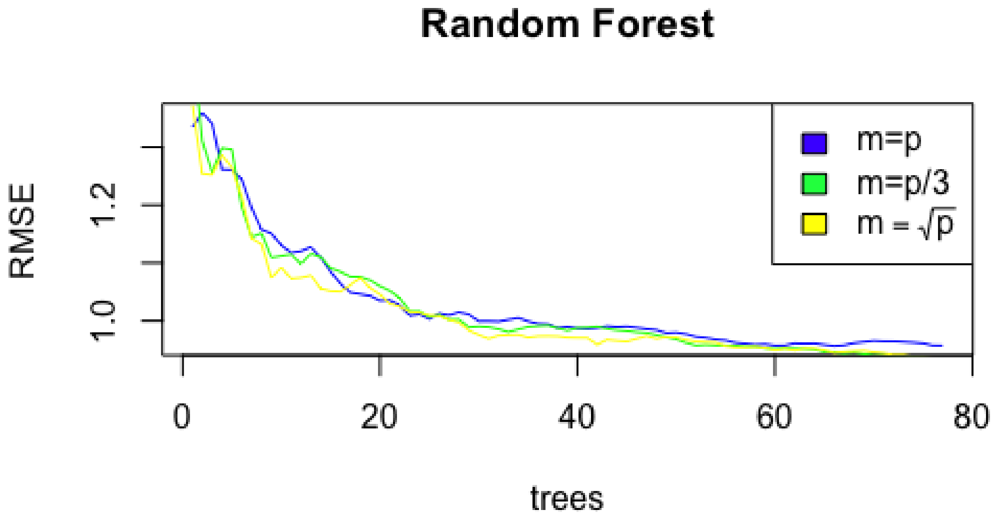
Figure D.0.1: Ridge and Lasso Coefficients



Note. This figure shows coefficient paths for the Ridge (to the left) and the Lasso (to the right) regression models on the L1-norm, $log(\lambda)$, and the fraction of deviance explained. The numbers on top of each figure are the non-zero regression coefficients.

Appendix E Random Forest Coefficients Candidates

Figure E.0.1: Random Forest: Number of candidates in each split



Note. Results from Random Forests with p exogenous variables. The error ($RMSE$) is displayed as a function of number of trees. We see from this plot that $m = p/3$ gives the lowest $RMSE$ estimate as number of trees increases. Our model chooses the number of trees that minimizes $RMSE$.

Appendix F Macroeconomic Variables

F.1 Transformation

$$X_{it} \text{ is the transformed series}$$

$$X_{it}^{A} \text{ is the untransformed series}$$

1. Code Δlv : $X_{i,t} = X_{i,t}^{A} - X_{i,t-1}^{A}$
2. Code $\Delta^2 lv$: $X_{i,t} = \Delta^2(X_{i,t}^{A})$
3. Code Δln : $X_{i,t} = ln(X_{i,t}^{A}) - ln(X_{i,t-1}^{A})$
4. Code $\Delta^2 ln$: $X_{i,t} = \Delta^2 ln(X_{i,t}^{A})$
5. Code $\Delta^3 ln$: $X_{i,t} = \Delta^3 ln(X_{i,t}^{A})$

Table F.1.1 discloses all data series used in the data sets and lists the transformation (Tran) applied to each variable to make them stationary. We have taken the logarithm of all variables with values > 0 and for all variables difference them until they reject (for KPSS – not reject) the stationary tests at a 99% confidence level. The Table also provides a short description of each variable with their corresponding short names. We have also listed their raw frequencies (freq.) to see whether they are daily (D), monthly (M) or quarterly (Q) published. All data series are gathered from Macrobond.

Table F.1.1: Transformation of Raw Data

No.	Short Name	Tran	Description	Freq.
1	NOGDPNOK	$\Delta^2 ln$	Norway, Gross Domestic Product (Mainland), Total, Constant Prices, SA, Market Prices, NOK	Q
2	NOOBXTR	Δln	Norway, Equity Indices, Oslo Stock Exchange, OBX Index (OBX), Total Return, Close, NOK	D
3	NOCSTR	Δln	Norway, Equity Indices, Oslo Stock Exchange, Communication Services, Index, Total Return, Close, NOK	D
4	NOBITR	Δln	Norway, Equity Indices, Oslo Stock Exchange, Benchmark Index, Total Return, Close, NOK	D
5	NOCDTR	Δln	Norway, Equity Indices, Oslo Stock Exchange, Consumer Discretionary, Index, Total Return, Close, NOK	D
6	NOUTR	$\Delta^2 ln$	Norway, Equity Indices, Oslo Stock Exchange, Utilities, Index, Total Return, Close, NOK	D
7	NOETR	Δln	Norway, Equity Indices, Oslo Stock Exchange, Energy, Index, Total Return, Close, NOK	D
8	NOFTR	Δln	Norway, Equity Indices, Oslo Stock Exchange, Financials, Index, Total Return, Close, NOK	D
9	NOHCTR	Δln	Norway, Equity Indices, Oslo Stock Exchange, Health Care, Index, Total Return, Close, NOK	D
10	NOITR	$\Delta^2 ln$	Norway, Equity Indices, Oslo Stock Exchange, Industrials, Index, Total Return, Close, NOK	D
11	NOITTR	Δln	Norway, Equity Indices, Oslo Stock Exchange, Information Technology, Index, Total Return, Close, NOK	D
12	NOCSEB	Δ	Norway, Consumer Surveys, Finance Norway, Expectations Barometer, Total, Trend Adjusted, SA, Index	Q
13	NODDASP	$\Delta^2 ln$	Norway, Credit Indicators, Domestic Debt (C2), All Sectors, Total, Transactions, 12-Month Growth, Percent	M
14	NODDHP	Δ	Norway, Credit Indicators, Domestic Debt (C2), Households, Total, Transactions, 12-Month Growth, Percent	M
15	NOKGBP	Δln	Norway, FX Spot Rates, Central Bank of Norway, NOK per GBP, Fixing	D
16	NOKEUR	Δln	Norway, FX Spot Rates, Central Bank of Norway, NOK per EUR, Fixing	D
17	NOKUSD	Δln	Norway, FX Spot Rates, Central Bank of Norway, NOK per USD, Fixing	D
18	NOEXCP	Δln	Norway, Expenditure Approach, Export, Total, Constant Prices, SA, NOK	Q
19	NOIMCP	Δln	Norway, Expenditure Approach, Import, Total, Constant Prices, SA, NOK	Q
20	NOGFCFCP	$\Delta^2 ln$	Norway, Expenditure Approach, Gross Fixed Capital Formation, Total, Constant Prices, SA, NOK	Q
21	NOSR10Y	$\Delta^2 ln$	Norway, Swap Rates, Macrobond, NOK, 10 Year, Mid	D
22	NOSR5Y	Δln	Norway, Swap Rates, Macrobond, NOK, 5 Year, Mid	D
23	NOSR2Y	Δln	Norway, Swap Rates, Macrobond, NOK, 2 Year, Mid	D
24	NOSR10Y2Y	Δ	Norway, Swap Rates, Macrobond, NOK, 10 Year, Mid minus Norway, Swap Rates, Macrobond, NOK, 2 Year, Mid	D
25	NIBOR3M	Δln	Norway, Interbank Rates, NIBOR, 3 Month, Fixing	D
26	NOMT	$\Delta^2 ln$	Norway, Manufacturing, Total, SA, Index	M
27	NOIPT	Δln	Norway, Industrial Production, Total, SA, Index	M
28	NOCUMWA	Δln	Norway, Capacity Utilization, Manufacturing, Weighted Average, SA	Q
29	NOCPI	Δln	Norway, Consumer Price Index, Total, Index	M
30	NOREPT	$\Delta^2 ln$	Norway, Real Estate Prices, All Residential Buildings, Total, SA, Index, Statistics Norway, Residential, Price Index	Q
31	NOCPIATE	$\Delta^3 ln$	Norway, Consumer Price Index, CPI-ATE, Index (CPI adjusted for tax changes and excluding energy products)	M
32	NOFCEHNPISHCP	Δln	Norway, Expenditure Approach, Final Consumption Expenditure, Households & NPISH, Total, Constant Prices, SA, NOK	Q
33	NOGFCFETPCP	Δln	Norway, Expenditure Approach, Gross Fixed Capital Formation, Extraction & Transport via Pipelines, Total, Constant Prices, SA, NOK	Q
34	NOGFCFMCP	$\Delta^2 ln$	Norway, Expenditure Approach, Gross Fixed Capital Formation, Mainland, Total, Constant Prices, SA, NOK	Q
35	NOFDMCP	$\Delta^2 ln$	Norway, Expenditure Approach, Final Demand from Mainland, Total Excluding Changes in Stocks, Constant Prices, SA, NOK	Q
36	NOCSTR	Δln	Norway, Equity Indices, Oslo Stock Exchange, Consumer Staples, Index, Total Return, Close, NOK	D
37	CDAXEUR	$\Delta^2 ln$	Germany, Equity Indices, Deutsche Boerse, DAX, CDAX Index, Price Return, Close, EUR	D
38	USBSM	Δln	United States, Business Surveys, ISM, Report on Business, Manufacturing, Purchasing Managers', SA, Index	M
39	USBSCEO	Δln	United States, Business Surveys, Conference Board, CEO Confidence Survey, Overall, Total, Measure of CEO Confidence, Index	Q
40	STOXXEUR	$\Delta^2 ln$	EU, Equity Indices, STOXX, Enlarged 15 TMI, Index, Net Total Return, Close, EUR	D
41	EUESESI	Δln	EU, Economic Surveys, DG ECFIN, Economic Sentiment Indicator, Balance, SA, Index	M
42	FTSEGBP	Δln	United Kingdom, Equity Indices, FTSE, All Cap, Index, Price Return, Close, GBP	D
43	OMXS50SEK	$\Delta^2 ln$	Sweden, Equity Indices, Nasdaq OMX, Benchmark, OMXS50 Equal Weighted Index, Total Return, Close, SEK	D
44	SEGDPSEK	Δln	Sweden, Gross Domestic Product, Total, Constant Prices, SA, Market Prices, SEK	Q

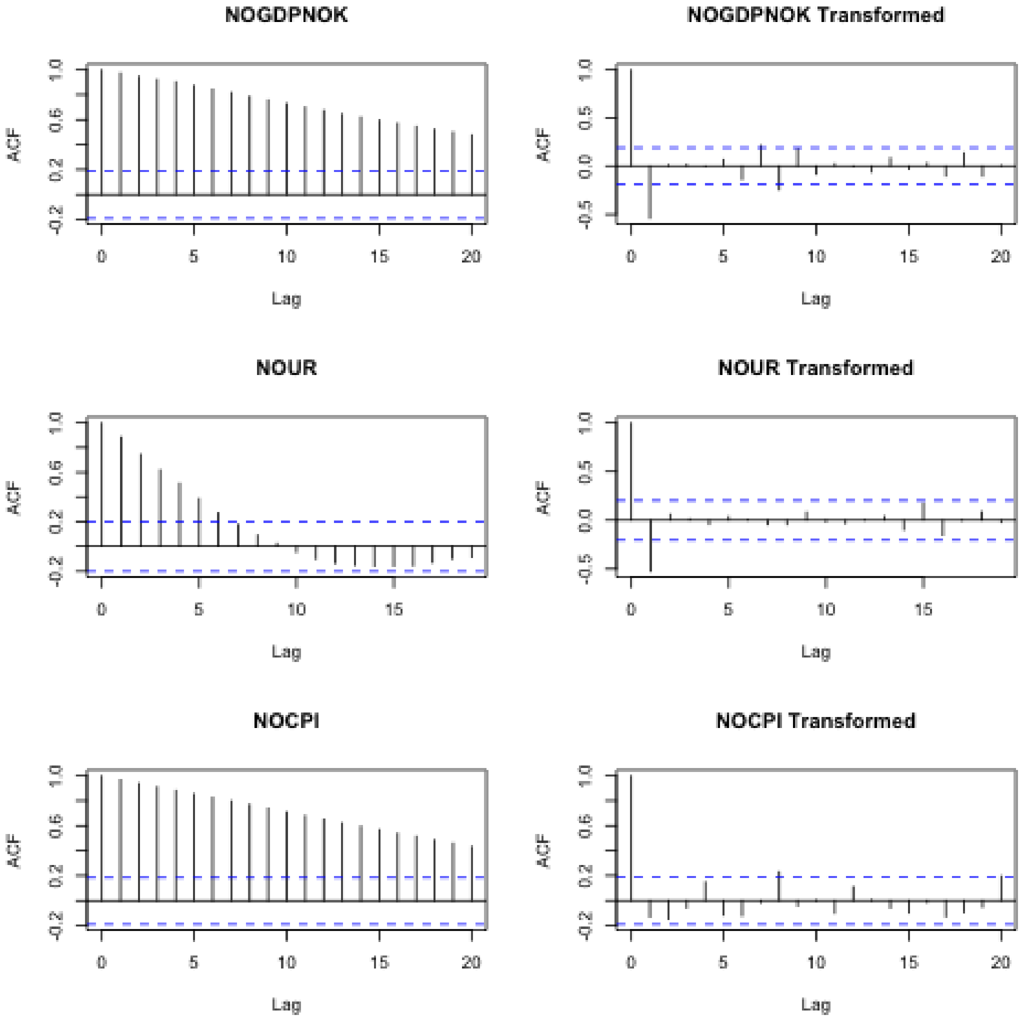
Continue...

… Continued

No.	Short Name	Tran	Description	Freq.
45	USGDPUSD	Δln	United States, Gross Domestic Product, Total, Constant Prices, SA, Chained, AR, USD	Q
46	DEGDPEUR	Δln	Germany, Gross Domestic Product, Total, Calendar Adjusted (X-13 ARIMA), Constant Prices, SA (X-13 ARIMA), Chained, EUR	Q
47	UKGDPGBP	$\Delta^2 ln$	United Kingdom, Gross Domestic Product, At Market Prices, Constant Prices, SA, GBP	Q
48	CNGDPCNY	$\Delta^2 ln$	China, Gross Domestic Product, National, Total (Official), Current Prices, CNY	Q
49	WRLDMP	Δln	World, Metal Production, Primary Aluminium, Total	M
50	NOUR	$\Delta^2 ln$	Norway, Unemployment, Rate, Males & Females, Total 15-74 Years, SA	Q
51	NOEP	Δln	Norway, Energy Production, Transmission & Distribution, Electricity Production, Total	M
52	UKPR	Δln	United Kingdom, Policy Rates, Bank Rate	D
53	EUHICP	$\Delta^2 ln$	EU, Consumer Price Index, All-Items HICP, Index	D
54	USGB2Y	Δln	United States, Government Benchmarks, Macrobond, 2 Year, Yield	D
55	USGB10Y	Δln	United States, Government Benchmarks, Macrobond, 10 Year, Yield	D
56	USGB10Y2Y	Δ^2	United States, Government Benchmarks, Macrobond, 10 Year- 2 Year, Yield	D
57	USPR	$\Delta^2 ln$	United States, Policy Rates, Effective Rates, Federal Funds Effective Rate	D
58	NOBAI	$\Delta^2 ln$	Norway, Bankruptcies, Total, All Industries	M
59	NOCI	$\Delta^2 ln$	Norway, Construction Indicators, Construction Production Index, Total, SA, Index, Statistics Norway, Index	Q
60	NOCSC	$\Delta^2 ln$	Norway, Construction Status, Number, Dwellings, National, Total, Statistics Norway, Completed, Overall	Q
61	NOCSP	Δln	Norway, Construction Status, Number, Dwellings, National, Total, Statistics Norway, Permits, Overall	Q
62	NOCSS	$\Delta^2 ln$	Norway, Construction Status, Number, Dwellings, National, Total, Statistics Norway, Starts, Overall	Q
63	NOBirths	$\Delta^2 ln$	Norway, Births, Total	Q
64	NODeaths	Δln	Norway, Deaths, Total	Q
65	NOEmigration	Δln	Norway, Emigration, Total	Q
66	NOImmigration	$\Delta^2 ln$	Norway, Immigration, Total	Q
67	NOProductivity	Δln	Norway, Productivity, Costs & Hours Worked, Actual Working Hours per Week, Males & Females, Total	Q
68	NOCCI	Δln	Norway, Construction Cost Index, Residential Buildings, Total, Index	M
69	NOExportTotal	Δln	Norway, Export Prices, Total, Index	Q
70	NOExportSalmonW	Δln	Norway, Export Prices, Salmon, Frozen, Weight	D
71	NOImportTotal	Δln	Norway, Import Prices, Total, Index	Q
72	NOCUWA	Δln	Norway, Capacity Utilization, Consumer Goods, Weighted Average, SA	Q
73	NOESLS	$\Delta^2 ln$	Norway, Economic Surveys, Bank of Norway (Norges Bank), Regional Network Report, Labour Supply, Aggregated	Q
74	NOESCC	$\Delta^2 ln$	Norway, Economic Surveys, Bank of Norway (Norges Bank), Regional Network Report, Capacity Constraints, Aggregated	Q
75	NOEAGFCF	Δln	Norway, Expenditure Approach, Gross Fixed Capital Formation, Mainland, General Government, Total, Constant Prices, SA, NOK	Q
76	EUManufPMI	Δln	Euro Area, Euro Area, Markit, Manufacturing PMI, SA	M
77	EUSPMI	Δln	Euro Area, Euro Area, Markit, Services PMI Business Activity Index, SA	M
78	EMPMI	Δln	Emerging Markets, Emerging Markets, Markit, Services PMI Future Activity Index, SA	M
79	EMManufPMI	Δln	Emerging Markets, Emerging Markets, Markit, Manufacturing PMI, SA	M
80	USBSM	Δln	United States, Business Surveys, ISM, Report on Business, Manufacturing, Purchasing Managers', SA, Index	M
81	USBSNonManuf	$\Delta^2 ln$	United States, Business Surveys, ISM, Report on Business, Non - Manufacturing, NMI/PMI, Purchasing Managers' Index, SA, Index	M
82	CNBAPMI	Δln	China, China, Markit, Services PMI Business Activity Index, SA	M
83	CNManufPMI	Δln	China, China, Markit, Manufacturing PMI, SA	M
84	NOESOGP3M	Δ^2	Norway, Economic Surveys, Bank of Norway (Norges Bank), Regional Network Report, Output Growth, Aggregated, Past 3 Months, National, Change Y/Y	M
85	NOESOGN6M	Δ	Norway, Economic Surveys, Bank of Norway (Norges Bank), Regional Network Report, Output Growth, Aggregated, Next 6 Months - Figures from Previous Rounds, National, Change Y/Y	M
86	NOESEGN3M	Δ^2	Norway, Economic Surveys, Bank of Norway (Norges Bank), Regional Network Report, Employment Growth, Aggregated, Next 3 Months - Figures from Previous Rounds	M
87	NOESAWG	Δ^2	Norway, Economic Surveys, Bank of Norway (Norges Bank), Regional Network Report, Annual Wage Growth, Aggregated, Index	M
88	NOESProfitability	Δ^2	Norway, Economic Surveys, Bank of Norway (Norges Bank), Regional Network Report, Profitability, Aggregated, Change Y/Y	M
89	NOAICOG	$\Delta^2 ln$	Norway, Investments, Oil & Gas Activity, Accrued Investment Costs, Extraction of Crude Oil & Natural Gas, Total, NOK	Q

Appendix G Auto Correlation Function

Figure G.0.1: Autocorrelation Function Plot

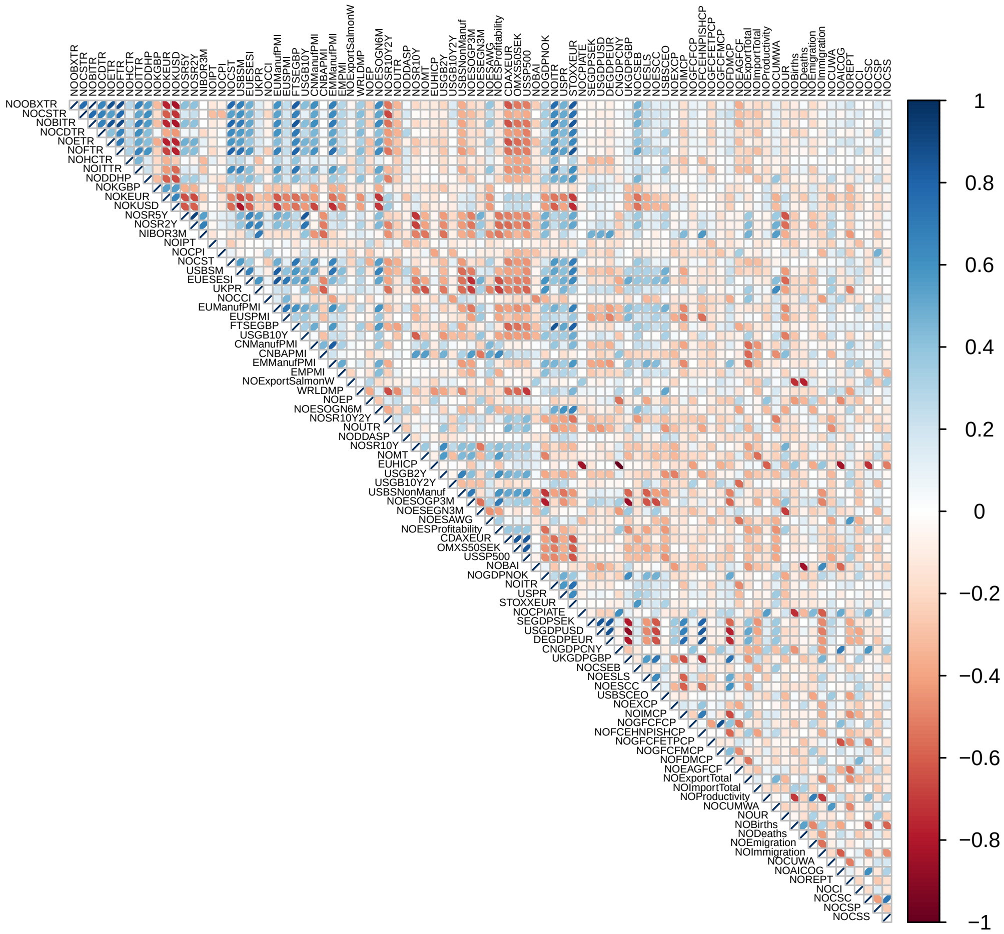


Note. This figure shows the properties of the three time-series Norwegian GDP, Unemployment Rate and Inflation. To the left we have the time-series autocorrelation plots before being transformed. To the right all three variables are transformed to be stationary. Along the x–axis we have the lagged values from 1–20 and their corresponding correlation with *Lag* 0. We consider values that rises above or falls below the dashed lines to be statistically significant.

Appendix H Correlation Matrix

H.1 Quarterly Data Set

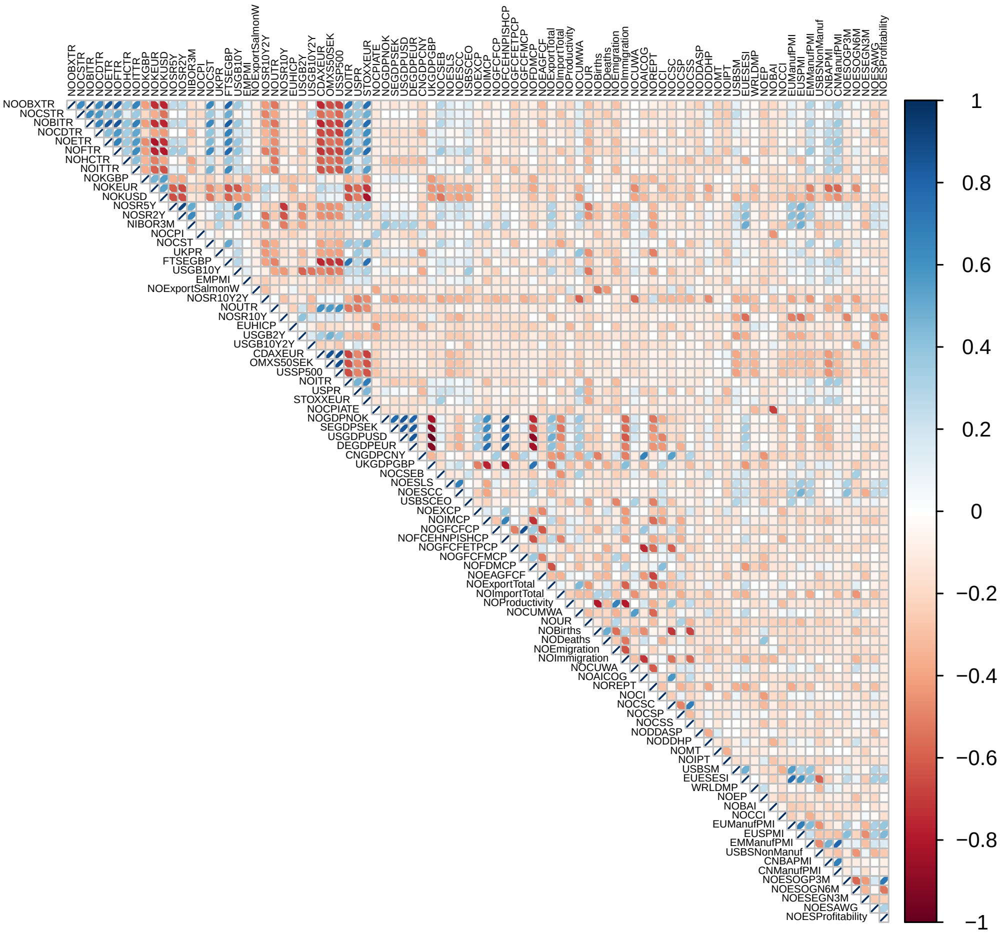
Figure H.1.1: Correlation Matrix: Quarterly Data Set



Note. This Figure shows the correlation between the time–series in the quarterly data set. A clear blue indicates correlation equal 1 hence the two variables are very correlated. A clear red indicates the opposite; two negatively correlated variables with correlation equal -1. White/weak color indicates little to no correlation.

H.2 Monthly Data Set

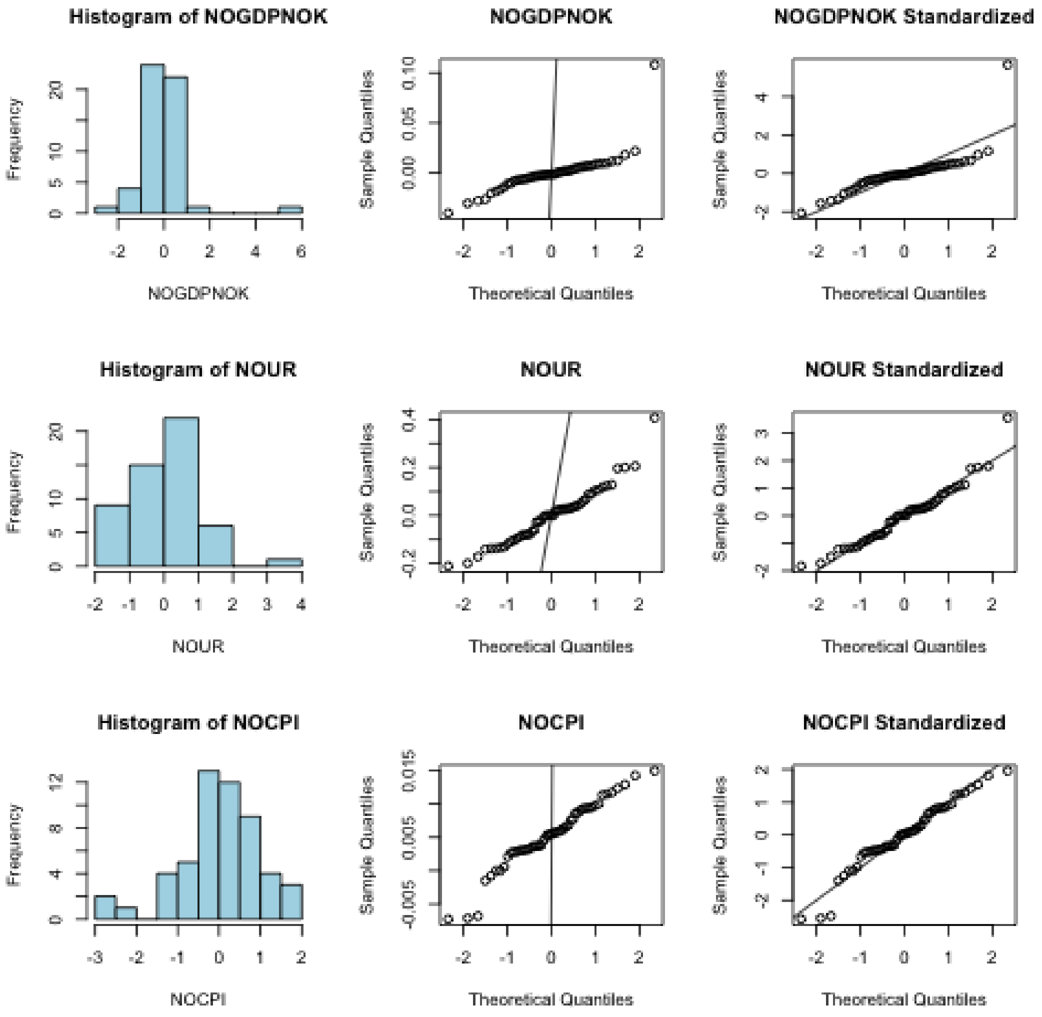
Figure H.2.1: Correlation Matrix: Monthly Data Set



Note. This Figure shows the correlation between the time–series in the monthly data set. A clear blue indicates correlation equal 1 hence the two variables are very correlated. A clear red indicates the opposite; two negatively correlated variables with correlation equal -1. White/weak color indicates little to no correlation.

Appendix I Time–Series Distribution

Figure I.0.1: Histogram and QQ–Plot



Note. This Figure plots the histogram and quantile–quantile plots of the Norwegian GDP, Unemployment Rate and Inflation both before and after being standardized. The thin line is to visualize the normal distribution with slope 1. Values deviation from this line is said to not fit with the normal distribution.

REFERENCES

Amadeo, K. 2020. Why Every Jobless Person Is Not Counted as Unemployed. https://www.thebalance.com/unemployment-rate. Section: The Balance.

AQR. Machine Learning: Why Finance Is Different. https://www.aqr.com/Learning-Center/Machine-Learning/Machine-Learning-Why-Finance-Is-Different.

Bankson, C. A., and A. M. Holm. 2019. Kunstig intelligens i makroøkonomisk prognosearbeid. *Norges Handelshøyskole* .

Bianchi, F., S. Ludvigson, and S. Ma. 2020. Belief Distortions and Macroeconomic Fluctuations. Tech. Rep. w27406, National Bureau of Economic Research, Cambridge, MA. doi:10.3386/w27406.

Brooks, C. 2014. *Introductory Econometrics for Finance*, vol. 4. Cambridge University Press.

Brownlee, J. 2016. Gentle Introduction to the Bias-Variance Trade-Off in Machine Learning. https://machinelearningmastery.com/gentle-introduction-to-the-bias-variance-trade-off-in-machine-learning.

Cerqueira, V., L. Torgo, and C. Soares. 2019. Machine Learning vs Statistical Methods for Time Series Forecasting: Size Matters. *arXiv:1909.13316 [cs, stat]* ArXiv: 1909.13316.

Chakraborty, C., and A. Joseph. 2017. Machine Learning at Central Banks. *SSRN Electronic Journal* doi:10.2139/ssrn.3031796.

DataFlair. 2018. Advantages and Disadvantages of Machine Learning Language. https://data-flair.training/blogs/advantages-and-disadvantages-of-machine-learning.

Flovik, V. 2018. How (not) to use Machine Learning for time series forecasting: Avoiding the pitfalls. https://towardsdatascience.com/how-not-to-use-machine-learning-for-time-series-forecasting-avoiding-the-pitfalls.

James, G., D. Witten, T. Hastie, and R. Tibshirani. 2013a. Linear Model Selection and Regularization. In G. James, D. Witten, T. Hastie, and R. Tibshirani (eds.), *An Introduction to Statistical Learning: with Applications in R*, Springer Texts in Statistics, pp. 203–264. New York, NY: Springer. doi:10.1007/978-1-4614-7138-7_6.

James, G., D. Witten, T. Hastie, and R. Tibshirani. 2013b. Tree-Based Methods. In G. James, D. Witten, T. Hastie, and R. Tibshirani (eds.), *An Introduction to Statistical Learning: with Applications in R*, Springer Texts in Statistics, pp. 303–335. New York, NY: Springer. doi:10.1007/978-1-4614-7138-7_8.

Jung, J.-K., M. Patnam, and A. Ter-Martirosyan. 2018. An Algorithmic Crystal Ball: Forecasts-based on Machine Learning. https://www.imf.org/en/Publications/WP/Issues/2018/11/01/An-Algorithmic-Crystal-Ball-Forecasts-based-on-Machine-Learning.

Kathuria, C. 2019. Regression — Why Mean Square Error? https://towardsdatascience.com/https-medium-com-chayankathuria-regression-why-mean-square-error.

Makridakis, S., E. Spiliotis, and V. Assimakopoulos. 2018. Statistical and Machine Learning forecasting methods: Concerns and ways forward. *PLOS ONE* 13:e0194889. doi:10.1371/journal.pone.0194889. Publisher: Public Library of Science.

Montgomery, A. L., V. Zarnowitz, R. S. Tsay, and G. C. Tiao. 1998. Forecasting the U.S. Unemployment Rate. *Journal of the American Statistical*

Association 93:478–493. doi:10.2307/2670094. Publisher: [American Statistical Association, Taylor & Francis, Ltd.].

Picardo, E. 2020. The Importance of GDP. https://www.investopedia.com/articles/investing/121213/gdp-and-its-importance.asp.

Singh, S. 2018. Understanding the Bias-Variance Tradeoff. https://towardsdatascience.com/understanding-the-bias-variance-tradeoff.

Stewart, M. 2020. The Limitations of Machine Learning. https://towardsdatascience.com/the-limitations-of-machine-learning.

Stock, J. H., and M. W. Watson. 1999. Forecasting inflation. *Journal of Monetary Economics* 44:293–335. doi:10.1016/S0304-3932(99)00027-6.

Svensson, L. E. O. 1997. Inflation forecast targeting: Implementing and monitoring inflation targets. *European Economic Review* p. 36.

Tiffin, A. 2016. Seeing in the Dark: A Machine-Learning Approach to Nowcasting in Lebanon. *IMF Working Papers* 16:1. doi:10.5089/9781513568089.001.

Varian, H. R. 2014. Big Data: New Tricks for Econometrics. *Journal of Economic Perspectives* 28:3–28. doi:10.1257/jep.28.2.3.