EC4308

Forecasting Monthly US GDP (INDPRO/ Industrial Production: Total Index)

Li Jingcheng (A0155214N)

Andrew Kong Fu Sheng (A0171839U)

Wong Chun Kitt (A0154205R)

Aw Zhen Yi (A0167312L)

Ang Khar Tsin (A0155647W)

1. Introduction

GDP is defined as the monetary value of all finished products and services produced within a country's borders within a specific time period. This includes anything produced within borders by foreigners and citizens alike. Since it is a representation of growth and production, GDP also serves as a useful indicator for the health of an economy (Kramer, 2017).

Considering its importance, it is no wonder that plenty of policies revolve around the state of the Economy. More importantly however, is that policy makers often have to rely on making real time forecasts with incomplete information on the current economic conditions. In addition, many key statistics are released with lags and also subjected to frequent revisions (Adam Richardson, 2018). This offsets the effectiveness of policies implemented due to inaccurate forecasts. Needless to say, GDP is also one of the many key statistics that policy makers wish they can better forecast.

1.1 Motivation

Although many of the concepts and algorithms behind machine learning methods were developed before the turn of the century, there were 3 major hurdles that prevented its application. First was the availability of data. Previously data collection was an arduous affair, which meant that there were not a lot of datasets to work with; especially if they were not made available to the public. The second reason was due to the lack of storage space. Even if the dataset was available, storage was an issue. Case in hand, IBM's first disk drive in 1980 only had a storage space of 2.5GB and cost $81,000. Lastly, poor computational power in general restricted any application of machine learning methods. Until GPUs started to replace CPUs, any form of training was extremely time intensive and inefficient as a result (Mishra, 2017).

However, with advances in technology, these issues rapidly became irrelevant. Machine learning methods were thus increasingly seen as viable alternatives when it comes to forecasting (Shaikh, 2017). It is also in this vein that our group decided on applying ML approaches on forecasting monthly GDP. There was a clear need and the approach can be considered sufficiently viable.

2. Literature Review

Before starting on the assignment, our group reviewed a number of prior studies that were done on GDP forecasting. This allowed us to narrow down the scope of our project and allowed us to validate the reliability of our results to an extent by cross referencing to other papers.

Habib Ahmed Elsayir weighed different autoregressive moving average (ARMA) models before concluding that the Autoregressive Integrated Moving Average (0,0,0) is the best model for predicting Sudan's annual GDP.

Massimiliano Marcellino's paper compared some of the common models that are used to benchmark growth and inflation.

Adam Richardson, Thomas van Florenstein Mulder and Tugrul Vehbi tried a range of popular machine learning algorithms to forecast New Zealand's quarterly GDP starting from the first quarter in 2009. These methods were in turn contrasted against benchmarks like an AR (1) model as well as various other statistical models used by the Reserve Bank of New Zealand. They concluded that ML algorithms outperformed these statistical benchmarks.

Lastly, we referenced the paper "*Using Monthly Indicators to Predict Quarterly GDP"* by Isabel Yi Zheng and James Rossiter for the dataset.

2.1 Existing benchmark models for predicting GDP

At present, there are many ways that GDP is forecasted. However, considering its seasonality and cyclical nature, time series has traditionally been the go to benchmark when it comes to forecasting GDP. This is reasonable because newer and more sophisticated models should at least be capable of outperforming models that only rely on the past behaviour of the variable of interest (Marcellino M. , 2005). The following models have been considered by various researchers when it comes to GDP forecasting.

$$y\_t = \alpha\_0 + \alpha\_1 y\_(t-1) + u\_t$$

This is an autoregressive model of order 1, with yt as the quarterly GDP growth to be forecasted, $\alpha 0$

and $\alpha 1$ as the parameters. ut is the residual term. Considering the simplicity and robustness (Marcellino A. B., 2002), this was used as the benchmark in one of the papers reviewed.

More complex models like Time-varying Auto regression (ARTV) and Artificial Neural Networks are also alternative benchmarks for capturing outlying observations and other forms of non-linearity induced by external events (Adam Richardson, 2018).

3. Data

The dataset used is obtained from FRED-MD by McCracken and Ng (2015) for the time period between January 1959 to August 2020 with a total of 740 monthly observations. The reasons for choosing FRED-MD includes its real-time update of the data through FRED database, being conveniently accessible as well as providing a rich source of data across different sectors of the economy. In relation to the aim of the project, which is to forecast the monthly GDP for US, it can be said that all of the variables in the dataset are relevant to GDP and hence, they were all used as predictors for forecasting. As there is no direct measure of GDP found in the dataset, the variable "Industrial Production (IP) index" is used as an approximation for GDP and the dependent variable of the forecast models. While the IP index is slightly different from GDP, it is an important macroeconomic indicator and fluctuations within the industrial sector can often account for most of the variation in economic growth

Customary examination of the dataset revealed missing values in some of the variables which were subsequently dropped. Following which, 12 lags of each variable were added to each observation. This is to increase the number of predictors and allow the machine learning models more options to select the best forecast model. The first twelve observations were then removed as they contain missing lagged variables. The observation for the same time period of each variable is also omitted as the forecast models will only use past data to predict GDP in the current time period. The total number of predictors summed up to 1404 after all the above-mentioned manipulation (Iyetomi et al., 2020).

4. Methodology

Throughout the paper, we will be selecting the models based on three information criterions. The first one is the Akaike Information criterion (AIC) which is generally used to select forecasting models. The second is AICc which is a version of AIC that has a correction for small sample sizes. The third will be the Bayesian Information criterion (BIC) which is generally used to select true models. A total of 6 machine

learning (ML) models were constructed to forecast the GDP of US as well as two benchmark models – random walk and Autoregressive (AR) models, to compare the performance between ML methods to traditional statistical ones. The ML methods include the Least Absolute Shrinkage and Selection Operator (LASSO), Post-LASSO (P-Lasso), Elastic Net, Random Forest (RF), a hybrid model between LASSO and RF and lastly an Ensemble method using simple average model between traditional and ML models.

As the dataset is a time series, the train/test set is split in a chronological order. The last 145 observations are reserved as the test set and all prior observations shall serve as the training set. A total of 4 forecast horizons – 1- 3-, 6- and 12-steps ahead forecasts were made across all models. The best models are selected based on the lowest test mean squared errors (MSE). Each of the ML models are compared to the two benchmark models to evaluate their performance based on the test MSEs. The Diebold-Mariano (DM) test will also be conducted to evaluate the models. A brief description of the above-mentioned ML methods is provided next.

4.1 Least Absolute Shrinkage and Selection Operator (LASSO)

Similar to ridge regression, LASSO shrinks the coefficients but the difference lies in the penalty term also known as the tuning parameter, λ. While ridge regression penalizes sum of squared coefficients, LASSO penalizes the sum of their absolute values. As a result, for high values of λ, LASSO can set some coefficients to zero thus performing variable selection while ridge regression is unable to.

The LASSO method are constructed with the *glmnet* function of the *glmnet* R statistical package and *rlasso* function of the *hdm* package . The first LASSO model uses the default tuning parameter, λ, denoted as grid in the *glmnet* function. The second model runs on user-defined grid that is selected based on AIC, BIC and AICc. The third model is rLASSO which uses plug-in rules for lambda. We used the default settings for *lasso* which assumes heteroskedasticity and independence. Finally, we compared the MSEs of each model with different tuning parameters to obtain the best LASSO model.

4.2 Post-LASSO (P-Lasso)

P-Lasso improves over the regular LASSO by reducing bias through applying the unpenalized ordinary least squares estimator (OLS) using only the variables selected by LASSO.

The P-Lasso method is constructed with the *glmnet* function of the *glmnet* R statistical package. The first P-Lasso model is constructed by running LASSO on a grid of lambdas, followed by applying OLS before choosing the best lambda with AIC, BIC and AICc. The second model differs slightly in that after running LASSO on a grid of lambdas, the best lambda is first chosen with the various ICs before applying OLS. Finally, compare the MSEs of the two models to obtain the best P-Lasso model.

4.3 Elastic Net

Elastic Net is a bridge between LASSO and ridge regression where it selects variables like LASSO and shrinks the coefficients of correlated predictors like ridge does. The method's advantage over LASSO and ridge regression lies in its computational efficiency and the ability to accommodate highly correlated variables unlike LASSO which has the tendency to drop variables which are highly correlated but yet important. This would prevent a possible loss in interpretability of the model.

The Elastic Net method is constructed with the *glmnet* function of the *glmnet* R statistical package. Similar to LASSO, the first model uses the default grid while the second model uses the user-defined grid.

4.4 Random Forest (RF)

RF is an ensemble method of decision tree regression that improves over Bootstrap Aggregating (Bagging) by forcing the algorithm to explore a richer set of models. While still taking a random subset of the training sample data as with Bagging, RF also takes a random number of features that is less than all the available features in the dataset. Doing so will ensure that the trees created are less correlated as compared to those in Bagging and thus increase the benefit of averaging the model's prediction. Coupled with the use of a fixed size rolling window, it allows random forest to be used to forecast time series models.

The RF method is constructed with the *randomForest* function of the *randomForest* R statistical package.

4.5 Hybrid (Lasso + RF)

The hybrid model presents an alternative to a typical ensemble by iterating different machines on the residuals from the previous machine. Doing so allows the model to take advantage of the strong features of different ML models to learn the aspects of the data that each model excels in.

The Hybrid method is constructed with the *rlasso* function of the *hdm* R statistical package, followed by the *tuneRF* function of the *randomForest* R statistical package. The hybrid model first runs LASSO and subsequently its residuals are used to fit random forest.

4.6 Ensemble (Simple Average between traditional and ML model)

The Ensemble method operates on the intuition that averaging over a combination of models' predictions takes advantage of each model's strengths thus giving rise to better predictions.

The Simple Average Ensemble method is constructed with the *lsei* function of the *lsei* R statistical package. The ensemble model takes one best model at each forecast horizon from both the traditional models and the ML models.

5. Results

Random Walk

The results from the random walk model are shown below:

Random Walk	MSE
1-step	0.6892966
3-step	2.653759
6-step	8.192238
12-step	23.85235

It can be seen that the 1-step forecast for the random walk model performs the best with the lowest MSE of 0.689

AR model

The results from the AR model are shown below:

AR	MSE
1-step	0.5911804
3-step	1.853526

6-step	6.063372
12-step	21.22212

It can be seen that the 1-step forecast for the AR model performs best with the lowest MSE of 0.5911804.

Lasso and P-Lasso

1-step forecast

We try LASSO using the default grid. The default consists of 80 points ranging from 0.613 to 23.1. Running Lasso using the default grid and selecting the best lambda using AIC, BIC and AICc gave us a test MSE of 1.051. We observe that the chosen lambda by ICs was relatively small and that values below 1 were insufficiently explored in the default grid. Therefore, we decided to use a user-defined grid which had 100 points between 0.001 and 1. Running Lasso using the lambda value selected by AIC and BIC gave us a test MSE of 0.699 while the lambda value selected by AICc gave us a test MSE of 4.17. Therefore we can observe that running Lasso using the user-defined grid and AIC/BIC performed the best.

Next, we tried running P-Lasso to see whether we can improve over the Lasso performance. We used the two P-LASSO models/ methods as stated in our methodology. In both methods, we used the user-defined grid. Using the first method, the lambda selected by AIC gave us a test MSE value of 0.863 while the lambda selected by BIC gave us a test MSE of 2.72. Using the second method, we simply used the lambda value which was selected previously in Lasso (AIC/BIC). This gave us a test MSE of 0.695. Therefore, we can observe that P-Lasso using the second method gave us a better performance. P-Lasso (AIC/BIC) also gave a slightly better performance as compared to Lasso (AIC/BIC).

Methods	**MSE**
P-Lasso (Method 2) - AIC/BIC	0.6953953
Lasso (user-defined) - AIC/BIC	0.6998594
P-Lasso (Method 1) - AIC	0.863655

rLasso	0.9445325
Lasso (default grid) - AIC/BIC/AICC	1.051000
P-Lasso (Method 1) - BIC	2.720331
Lasso (user-defined) - AICC	4.179424

In summary, we can observe that Lasso (user-defined grid)-AIC/BIC and P-Lasso (Method 2)-AIC/BIC gave the best performance. However, both were not able to beat the Random Walk and AR models. rLASSO also performed relatively well, as is the case for longer forecast horizons.

3-step forecast results

Methods	**MSE**
Lasso (user-defined grid) - AIC/BIC	2.325550*
Lasso (Default grid) - AIC/BIC/AICc	2.450993*
rLasso	3.088726
P-Lasso (Method 2) - AIC/BIC	9.752206
P-Lasso (Method 1) - BIC	15.85608
P-Lasso (Method 1) - AIC	40.25960
Lasso (user-defined) - AICc	303.9785

* - Able to beat the Random Walk model.

** - Able to beat both Random Walk and AR models.

6-step forecast results

Methods	**MSE**

rLasso	7.07756*
Lasso (default grid) - AIC/BIC/AICc	8.156728*
Lasso (user-defined) - AIC/BIC	9.255277
P-Lasso (Method 2) - AIC/BIC	20.82496
P-Lasso (Method 1) - BIC	32.08215
Lasso (user-defined grid) - AICC	262.1239
P-Lasso (Method 1) - AIC	6389.216

12-step forecast results

Methods	**MSE**
Lasso (user-defined) - AIC/BIC	19.66925**
Lasso (default grid) - AIC/BIC/AICc	21.15767**
rLasso	25.77846
P-Lasso (Method 2) - AIC/BIC	117.7721
P-Lasso (Method 1) - BIC	156.9411
Lasso (user-defined) - AICC	18266.93
P-Lasso (Method 1) - AIC	44819.17

From our 3,6 and 12 step forecast results above, we can observe that LASSO performs the best while P-LASSO performs poorly at longer forecast horizons. P-LASSO (AIC) is especially poor which may be due to AIC not being suited for high dimensional settings. LASSO was able to beat the Random Walk model but not the AR model at 3 and 6 step forecasts. LASSO was able to beat both Random Walk and AR

models at 12-step forecast. Therefore, we can conclude that the longer the forecast horizon, the better LASSO performs against Random Walk and AR models.

Variable selection in LASSO

In 1-step forecast, LASSO selected 6 predictors - the first lags of Real Manufacturing and Trade Industries Sales, Industrial Production Index, Industrial Production: Consumer Goods, Industrial Production: Durable Consumer Goods, Average Weekly Overtime Hours: Manufacturing and the second lag of Industrial Production: Consumer Goods. All the predictors selected are directly related to industrial production.

As the forecast horizon increases, the number of variables selected increases from 6 in 1-step forecast to 37 in 12-step forecast. More variables that are not directly related to industrial production are selected including those pertaining to Housing starts, S&P 500 performance, Interest rates, Medical care prices etc.

Elastic Net

Similar to our approach to the LASSO model, we try Elastic net using the default grid. We selected the best lambda using three measures (AIC/BIC/AICc) and obtained a test MSE of 6.986.

We also observe that the chosen lambda by ICs was relatively small and that values below 1 were insufficiently explored in the default grid. Therefore, we decided to use the user-defined grid which had 100 points between 0.001 and 1. Similar to before, we selected the best lambda from this grid using AIC, BIC and AICC. This time, AIC and BIC selected the same lambda which gave a test MSE of 2.608419. This performs better than the default grid. AICc selected a different lambda which resulted in a higher test MSE of 7.02109.

1-step forecast results

Methods	**MSE**
Elastic net (default grid) - AIC/BIC/AICC	6.985871
Elastic net (user-defined grid) - AIC/BIC	2.608419
Elastic net (user-defined grid) - AIC/BIC/AICC	7.021091

At this point, we decide to continue using the same methodology and generate the values of MSE for the 3,6 and 12-step forecasts.

For the 3 step forecast, all three measures (AIC/BIC/AICc) select the same lambda for the default grid and result in a test MSE of 8.9044.

As expected, the further forecasts tend to perform worse with the AIC and BIC being the measures that produce the lowest test MSEs in general.

3-step forecast results

Methods	**MSE**
Elastic net (default grid) - AIC/BIC/AICC	8.904435
Elastic net (user-defined grid) - AIC/BIC	3.33336
Elastic net (user-defined grid) - AIC/BIC/AICC	96.34134

6-step forecast

Methods	**MSE**
Elastic net (default grid) - AIC/BIC/AICC	10.03273
Elastic net (user-defined grid) - AIC	9.873392
Elastic net (user-defined grid) - BIC	11.09483
Elastic net (user-defined grid) - AICC	159.4677

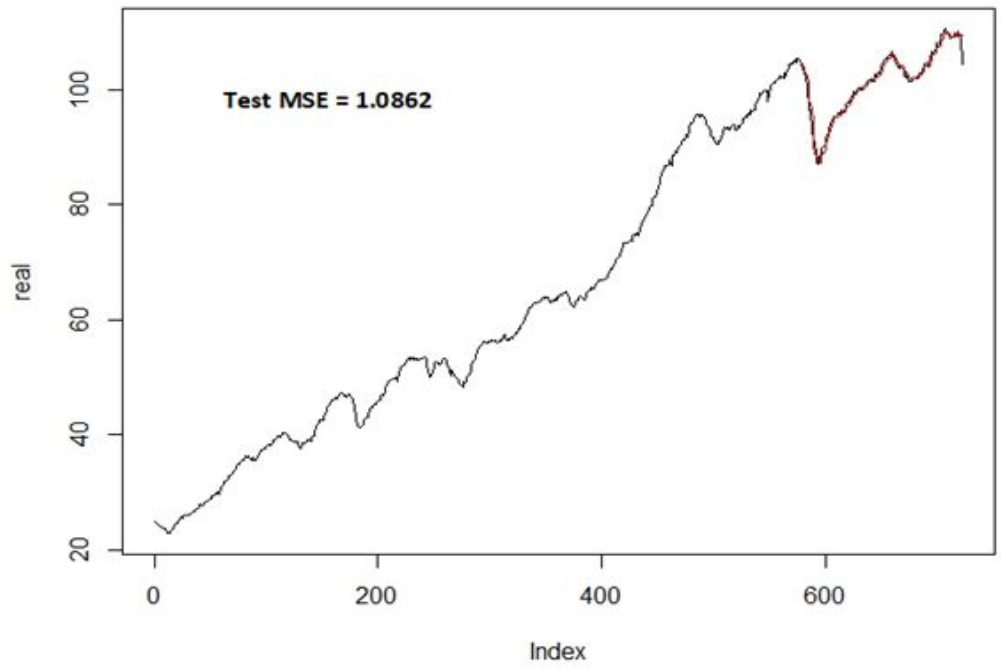
12-step forecast

Methods	**MSE**

Elastic net (default grid) - AIC/BIC/AICC	20.4249
Elastic net (user-defined grid) - AIC/BIC	23.51791
Elastic net (user-defined grid) - AICC	9106.187

Random Forest

We run randomForest() with mostly default settings, setting mtry to 1/3 of our predictors, which is 468. This process repeats for each rolling window, which then gives us our 145 predictions to compare against the test set to get our test mean squared error (MSE). For the 1 step forecast, the test MSE is computed to be around 1.0862, which is actually pretty small. The plot below shows that the predicted values (denoted by the red line) actually models the actual values closely, barring the sharp drop at the end for the actual values that the predictions were not able to catch.



Scanning across the board for the important variables, we found that variables such as the first and second lag of INDPRO consistently showed up with a high %IncMSE coefficient. This seems rather sensible and intuitional as we would expect the past data of our predicted variable to play a large part in the forecast.

We then run the similar steps for the to obtain the 3 steps ahead, 6 step ahead and 12 steps ahead forecast. Note that the size of each rolling window gets shrunk by h-1, for any h-step forecast to preserve the integrity of having the same test set of last 145 observations.

	Test MSE
1 step forecast	1.0862
3 step forecast	3.4521
6 step forecast	9.5457
12 step forecast	25.2061

The test MSE expectedly increases as the forecast horizon increases. Therefore, random forest may not be a very good technique to use to predict more than t+1 forecast because of the need to feed latest data back into the model as training data.

Hybrid and Ensemble methods

Hybrid learning

We first ran Lasso to account for the linear trend and then model the deviations using Random Forest. The results are below.

	Random Forest	**rLasso**	**Hybrid (rLasso + RF)**
1-step	1.0862725	0.9445325	**0.7897661**
3-step	3.452115	3.088726	3.094477
6-step	9.545745	7.07756*	**6.4963***
12-step	25.20615	25.77846	29.18848

We can observe from the table above that the Hybrid model was able to improve upon Random Forest and rLasso for 1 and 6-step forecasts. Overall the Hybrid model performed relatively poorly against benchmark models, only beating the Random Walk model at the 6-step forecast.

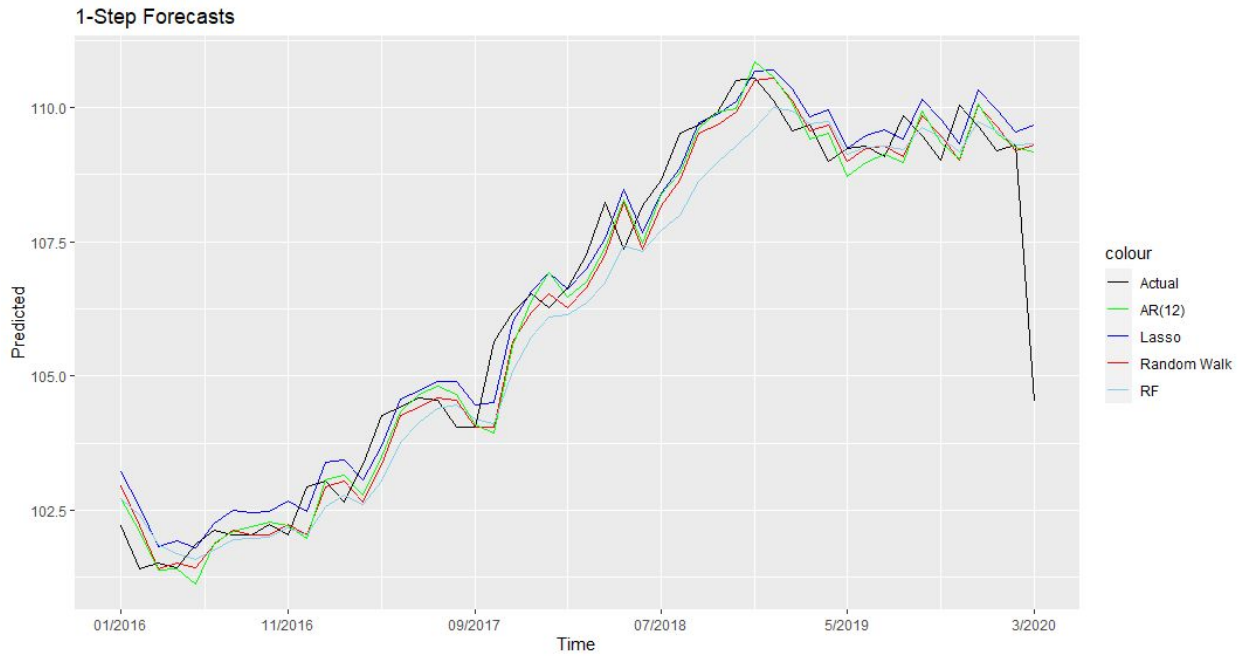
Ensemble method (AR+ML)

We took an average of the AR model and the top performing ML model. We hope to effectively harness the predictive power of both traditional and ML models. For 1-step forecast, we took the average of AR and P-Lasso predictions. For 3 and 12-step forecasts, we used Lasso and AR. For 6-step forecast, we used the Hybrid model and AR. The performance are as follows:

	Lasso (AIC/BIC)	P-Lasso (AIC/BIC)	Random Forest	Hybrid (rLasso + RF)	Random Walk	AR(12)	Ensemble (AR +ML)
1-step	0.6998594	0.6953953	1.0862725	0.7897661	0.6892966	**0.5911804**	0.613949
3-step	2.32555	9.752206*	3.452115	3.094477	2.653759	**1.853526**	1.923141
6-step	8.156728	20.82496*	9.545745	6.4963	8.192238	6.063372	**5.74955**
12-step	19.66925	117.7721	25.20615	29.18848	23.85235	21.22212	**18.71736**

We can observe that the Ensemble model can beat Random Walk models in all cases. The Ensemble model can also beat the AR(12) model at 6 and 12-step forecasts.

Illustrations




3-Step Forecasts
110.0
107.5
105.0
102.5
Predicted
01/2016
11/2016
09/2017
07/2018
5/2019
3/2020
Time
colour
Actual
AR(12)
Lasso
Random Walk
RF

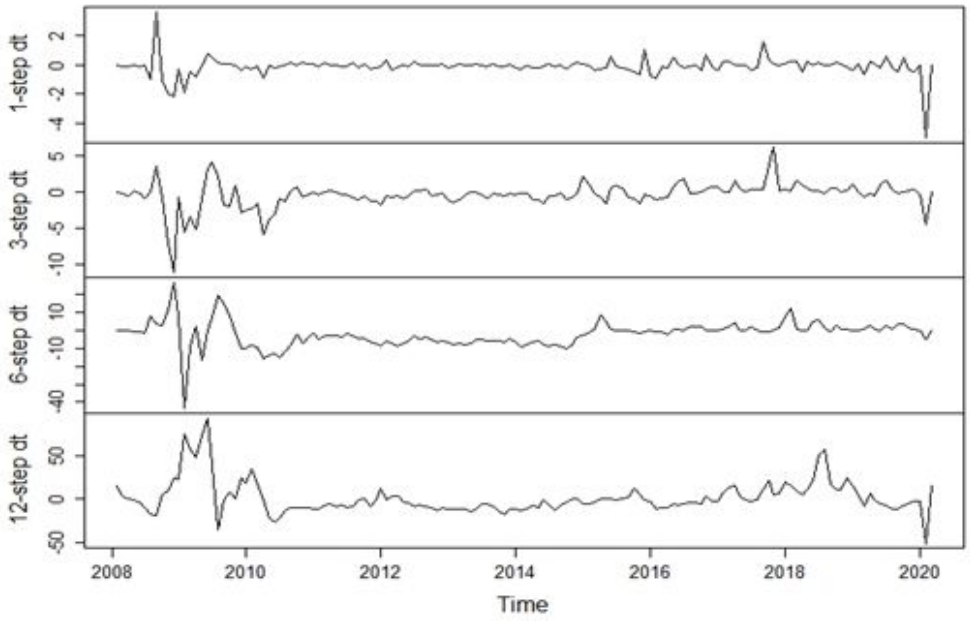

6-Step Forecasts
110.0
107.5
105.0
102.5
Predicted
01/2016
11/2016
09/2017
07/2018
5/2019
3/2020
Time
colour
Actual
AR(12)
Ensemble
Random Walk

The plots above illustrate the predictions of our models. We can observe that all the selected models perform relatively well at 1 and 3 step forecasts. The Random Walk model performs progressively worse at longer forecast horizons. We can observe that the Ensemble method performs very well as compared to AR and Random Walk at 6 and 12-step.

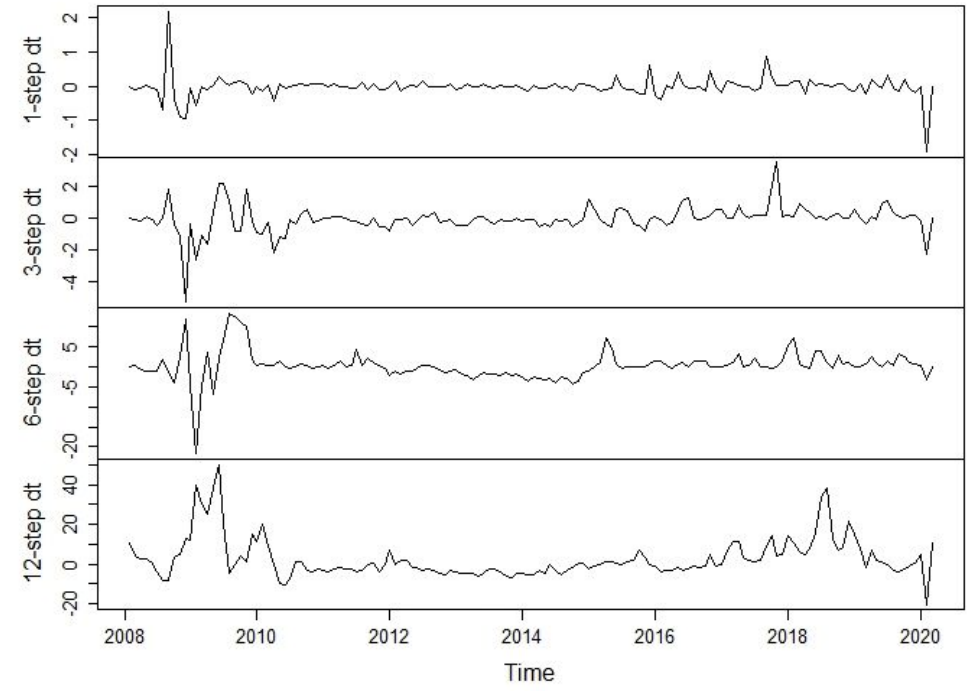
Diebold-Mariano (DM) Test

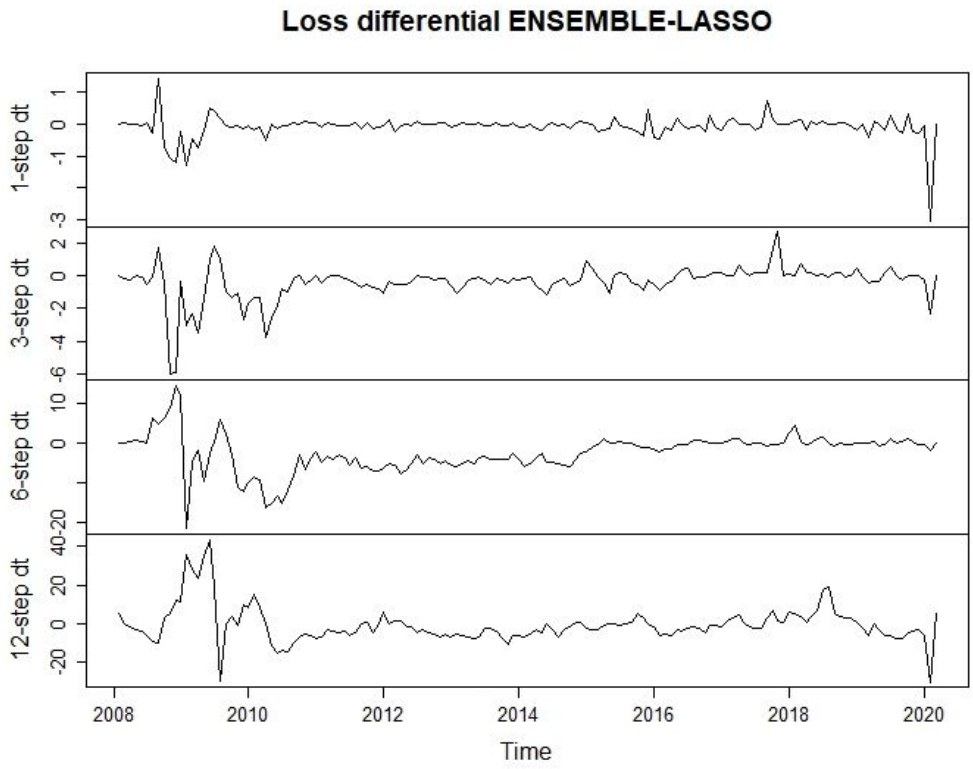
From the results of the test mean squared error that we have obtained, we observed that the benchmark AR model generally gives the lowest test mean squared error. However, some methods such as lasso, random forest can be quite comparable to it for certain forecasts. We can then test the predictive ability of these methods using the Diebold-Moriano(DM) test. We compute the loss differentials of AR-LASSO, RF-LASSO to see if it violates the stationary assumption of the DM test.

Loss differential AR-LASSO



Loss differential AR-ENSEMBLE





From the loss differential plots, we assumed that the stationary assumption holds as the line is relatively stable barring the sudden spikes and troughs at around 2009. However, it quickly stabilises and does not seem to affect the periods beyond it.

We proceed on to getting the DM test t-statistic by using the Newey-West estimator with 5 lags, as determined by taking the cube root on the number of predictors. The results obtained are shown below.

	AR-ENSEMBLE	**AR-LASSO**	**ENSEMBLE-LASSO**
1 step forecast	-1.0995	-1.9259	-2.2230
3 steps forecast	-0.2065	-1.6890	-2.3114
6 steps forecast	0.5875	-1.9014	-2.6324

12 steps forecast	1.0457	0.3400	-0.4517

The t-statistics for AR-ENSEMBLE, AR-LASSO are mostly negative, which corresponds to the fact that the AR methods having a lower test MSE as compared to the ensemble and lasso methods. We then check if this t-stat is statistically significant. Comparing AR and ensemble methods, we see that AR beats out ensemble for 1,3 steps forecast and ensemble beats out AR in the 6,12 steps forecasts. However, none of the forecasts are statistically significant. Therefore, we cannot conclude that both methods have different predictive abilities from each other. Comparing AR-LASSO, none of the forecasts are statistically significant at the 5% level, although they are at the 10% level for 1,3,6 steps ahead forecast. Comparing our best two methods, ensemble and lasso, we found that ensemble is better than lasso for all forecast horizons and are all statistically significant at the 5% level. We can conclude that the ensemble method indeed has different predictive ability as compared to lasso.

6. Conclusion

Overall, our paper explored 6 different machine learning methods to find out which method is the most suitable for our proposed problem. Based on the test mean squared error alone, we found that benchmark models like Random Walk and AR models perform relatively well at 1 and 3-step forecasts while ML methods perform well at 6 and 12-step forecasts. In particular, Lasso was able to beat Random Walk and AR for 3,6,12-step and 12-step respectively. It is interesting that ML methods tend to perform better for longer range forecasts.

The Hybrid model which combines Lasso and Random Forest performed well but was unable to beat the AR model. Since Lasso performs well but not Random Forest and the Hybrid model, we can infer that the relationship between our dependent and independent variables is generally linear.

The ensemble method, which combines AR with 1 best performing ML model, produced the best results for 6 and 12 step forecasts. The ensemble method allows us to combine the predictive qualities of our X variables lags on top of Y variable lags used in the AR model.

Looking at our DM test results, we are unable to say that ML methods are better than the benchmark models at the 6 and 12-step. In our research paper, we explored various ML methods but these methods

were not exhaustive. A future area of research could be further exploration of tree-based methods as well as ensemble methods such as Granger-Ramanathan combination and Lasso combination.

7. Reference

Adam Richardson, T. v. (2018). *Nowcasting New Zealand GDP using machine learning algorithms.*

Elsayir, H. A. (2018). An Econometric Time Series GDP Model Analysis: Statistical Evidences and Investigations . 15.

Iyetomi, H., Aoyama, H., Fujiwara, Y., Souma, W., Vodenska, I., & Yoshikawa, H. (2020). *Relationship between Macroeconomic Indicators and Economic Cycles in U.S.* Scientific reports, *10*(1), 8420. https://doi.org/10.1038/s41598-020-65002-3

Kramer, L. (2017, October 17). *What Is GDP and Why Is It So Important to Economists and Investors?* Retrieved from Investopedia : https://www.investopedia.com/ask/answers/what-is-gdp-why-its-important-to-economists-investors/

Marcellino, A. B. (2002). Are there any reliable leading indicators for US Inflation and GDP Growth? 45.
Marcellino, M. (2005). *A benchmark for models of growth and inflation.*

Mishra, A. (2017, July 21). *Quora* . Retrieved from https://www.quora.com/Why-couldnt-we-use-AI-and-machine-learning-before

Rossiter, I. Y. (2006). *Using Monthly Indicators to PredictQuarterly GDP.* Bank of Canada .

Shaikh, F. (2017, May 18). *Why are GPUs necessary for training Deep Learning models?* Retrieved from Analytics Vidhya: https://www.analyticsvidhya.com/blog/2017/05/gpus-necessary-for-deep-learning/