

# ***Forecasting US GDP in the Post-pandemic Age: An Evaluation of AR Model and Machine Learning Algorithms***

**Yifeng Jin**

*College of Arts and Sciences, Emory University, Atlanta, United States  
yifeng.jin@emory.edu*

**Abstract:** The AR (1) model outperforms the random forest, KNN, gradient boosting, and Xgboost algorithms by yielding a lower MAE in the prediction result. The reason behind may be attributed to AR model's ability to utilize data after a sudden shock. Since AR model predict one years' GDP based on the previous year's GDP, after a sudden shock, the effect of the shock will be captured when predicting the GDP of the year after the shock. But the machine learning algorithms evaluated in this research are trained using data in a fixed time frame. They are not able to utilize data in the testing data set and adjust accordingly. By evaluating the performance of the models, this research reveals the importance of a model's ability to handle information of a sudden shock in GDP forecasting and that AR model possess such ability while random forest, KNN, gradient boosting, and Xgboost do not.

**Keywords:** GDP forecasting, Machine learning, Autoregressive Model, Covid-19

## **1. Introduction**

Although the covid-19 pandemic has ended for over 3 years, there are still lingering impacts that continues to influence our economy today, especially the forecasting of GDP. The pandemic was a sudden shock to all economies around the globe and put many of them into a recession. The unemployment rate surged and production lines were disrupted [1]. Then, some governments implemented economic stimulating policies that caused a prolonged period of inflation. The high inflation in the US continued to 2024 and can still be partly attributed to the expansionary fiscal and monetary policies conducted during the pandemic [2].

All techniques of forecasting GDP requires the construction of a model using large amount of past economic data. But since the pandemic has substantially impacted our economy, the credibility of models trained using pre-pandemic data becomes questionable [3]. However, it is also infeasible to use solely post-pandemic data. It is difficult for machine learning algorithms to produce credible results using only four years of data.

Therefore, in this paper, we are going to evaluate the performance of different models trained using pre-pandemic data on forecasting post-pandemic US GDP. This will not only provide insights into the performance of those algorithms but also an evaluation of the impact of the pandemic as well as policies in response to it on US GDP [4]. The models used include Autoregressive model, Random Forest, K-Nearest Neighbors, Gradient Boosting, and XGBoost.

## 2. Previous Researches

Plenty of previous works have been done on evaluating the performance of traditional economic forecasting techniques, namely AR models, and machine learning algorithms on forecasting GDP. However, the results are not consistent across different researches.

In a research [5] that compares the predictive power of AR and Long-short memory model (LSTM) on GDP and consumer price of G7 countries, Yutaka and Akio found that AR model performed better than LSTM in forecasting GDP. They used data from 1992Q1 to 2017Q3 to forecast the data in 2017Q4. They used only one variable to construct the AR and LSTM model: past GDP for prediction of GDP and past inflation rate to predict inflation rate. The root mean squared error calculated indicates that AR model outperforms LSTM when forecasting GDP. But there is no significant difference in performance when forecasting inflation rate.

While in a research [6] that compares KNN with various traditional time series analysis approach including AR, ARX, SARIMA, SARIMAX, and linear regression, Giovanni, Giacomo, and Sara found that KNN outperforms the other models. They used quarterly data of real US GDP from 1976 to 2020 to conduct both one quarter ahead and multiple quarters ahead forecast of the GDP from first quarter of 2019 to the last quarter of 2020. Unlike Yutaka and Akio [5], they incorporated multiple relevant variables in their model. The variables are macroeconomic indicators like interest rate and CPI. The researchers found that GDP has strong self-explanatory power – that is, GDP in one quarter and explain the GDP in the next quarter well – and take advantage of it by introducing the traditional time series analysis approaches. Also, they introduced multiple variables to take advantage of the computing power of KNN model.

## 3. Methodology and Data

Although the forecasting scenarios and data used are different in the previous researches, we can see that AR and machine learning models has different advantages. AR model is a linear model that relies solely on past data of a variable to predict itself. Therefore, it is very suitable for predicting variable that have strong self-explanatory power like the GDP [7]. Also, the predictors selected for such time series analysis should also possess such quality for better prediction. [8] On the other hand, machine learning algorithms can handle multiple variables and interpret the various non-linear and even non-parametric relationships between them easily [9]. To fully utilize the predictive power of machine learning algorithms, this research will introduce a large variety of macroeconomic indicators to forecast GDP.

In this research, we will use the following data of the United States from 1980 to 2023: real GDP, unemployment, inflation, PPI, house price index, total population, total stock market value, one year treasury bond rate, credit spread, net export, personal savings rate, and the average PE ratio of S&P 500 stocks. The total stock market value is collected from the World Bank Open Data and the rest are all collected from the Federal Bank of St. Louis.

The data before covid-19 outbreak, which are from 1980 to 2019, will be used to train the models and the data after covid-19 outbreak, which are from 2020 to 2023, will be in the testing set to evaluate the performance of the models.

### 3.1. AR model

An autoregressive model is a linear regression of one year's data on its values in the past years [10]. The number of past values incorporated in the regression is the order of AR model. In this research, the AR model of order one is used, meaning that each year's real GDP is the dependent variable and the real GDP of the year before each year is the independent variable. The formula is as follows:

$$Y_t = \beta Y_{t-1} + \varepsilon$$

The  $\beta$  is the regression coefficient and  $\varepsilon$  is the error term that captures all the noise in the data. This model is trained in python using the statsmodel package.

### 3.2. Random Forest

Random forest is an extension of the decision tree algorithm. A decision tree splits the dataset based on a binary condition of the data points' features recursively. For each split, among all possible conditions, the condition that decreases the variance of the dataset after split the most is selected. After a certain number of split or when the variance is under a desired threshold, the decision tree stops splitting. For any new data point, the prediction will be the mean of the datapoints that satisfy the same binary conditions of the decision tree [11].

A random forest algorithm creates multiple decision trees and aggregate the result of the trees to obtain the final result. It randomly selects a certain number of observations and a certain number of features or variables from the original dataset and then train a decision tree based on each selection. The aggregation of multiple decision trees based on randomly sampled data makes the model less sensitive to subtle changes in the training data.

This model is trained in python using the sklearn package.

### 3.3. K-Nearest Neighbors (KNN)

The KNN model determines the value of a data point based on the nearest  $k$  data points. The algorithm clusters the data points based on the value of the independent variables. Then, the value of the dependent variable is predicted as the mean of the  $k$  nearest values. This model is trained in python using the sklearn package.

### 3.4. Gradient Boosting

Gradient boosting is also an extension of the decision tree algorithm. It creates multiple decision trees and aggregate the results to obtain the final result. But unlike random forest, every decision tree gradient boosting creates depends on the results of the previous tree. The learning process starts from simply taking the mean of the dependent variable in the training set. Then, the residual error for each observation is calculated and a new decision tree is built to split the data until a certain number of leaves, which is preset as a parameter, is created. As the number of leaves is restricted, if more than one value is on the same leaf, the value on that leaf will be replaced with their mean [12].

The predictions of residuals obtained by a decision tree will be scaled by a factor between 0 to 1, named learning rate, to prevent the model from over-fitting to the training data. And the actual prediction of this new tree will be the prediction of the previous tree plus the product of learning rate and the predicted residuals from the new tree. Then, the new residual error is calculated and the same process is repeated until a certain number of decision trees, which is also a preset parameter, is created. The final result  $\hat{Y}$  of gradient boosting is as follows:

$$\hat{Y} = \bar{Y} + \gamma r_1 + \gamma r_2 + \dots + \gamma r_n \quad (1)$$

Where  $\bar{Y}$  is the mean of training data,  $r_i$  is the residual error predicted by each decision tree, and  $\gamma$  is the learning rate.

### 3.5. XGBoost

XGBoost is an extension of the Gradient Boosting algorithm. Similar to Gradient Boosting, it creates multiple trees to predict the residual error made by the past prediction and aggregate them to produce the final result. The differences between the two algorithms are the condition of splitting data and create branches in the decision tree, the inclusion of a regularization parameter, and the inclusion of a pruning process [13].

In Gradient Boosting, the condition to split data at each node in the decision tree is the one that decreases the variance of the data the most after split. While in XGBoost, the condition is the one that increases the similarity score of the data the most after split. The algorithm stops splitting the data after a certain number of times, which is a preset parameter. The similarity score of a leaf is calculated as:

$$\text{similarity score} = \frac{(\sum \text{residual})^2}{\text{number of residual} + \lambda} \quad (1)$$

Where  $\lambda$  is a preset regularization parameter, ranging from 0 to positive infinity, that prevents the model from overfitting to the training data. After the split, the condition that maximizes the increase in similarity score is chosen. This parameter is also included when calculating the prediction result of each leaf:

$$\text{predicted residual} = \frac{\sum \text{residual}}{\text{number of residual} + \lambda} \quad (2)$$

Before the algorithm move on to build the next tree, XGBoost has an additional step of pruning. Starting from the last leaf created in the decision tree, if the increase in similarity score is smaller than gamma, which is a preset parameter, that split will be pruned, or eliminated from the decision tree. The process continues to the second last leaf in the decision tree until the increase in similarity score of one leaf is larger than gamma.

Then, next tree is built based on the residual error of the prediction of this tree until a certain number of trees has been created. The final result is also calculated the same way as in Gradient Boosting: summing the base of prediction and all the products of predicted residuals and a learning rate.

This model is trained in python using the xgboost package.

### 3.6. Evaluation of Model Performance

The performance of a model is how close to reality is the prediction made by a model. This is measured by the mean absolute error (MAE): the average absolute difference between a model's prediction and the actual real GDP in the testing data set. The formula is as follows:

$$MAE = \frac{1}{n} \sum |Y_{\text{actual}} - Y_{\text{predicted}}| \quad (1)$$

The mean absolute error will be presented as a percentage of the average real GDP of the testing data set.

### 3.7. Data Selection

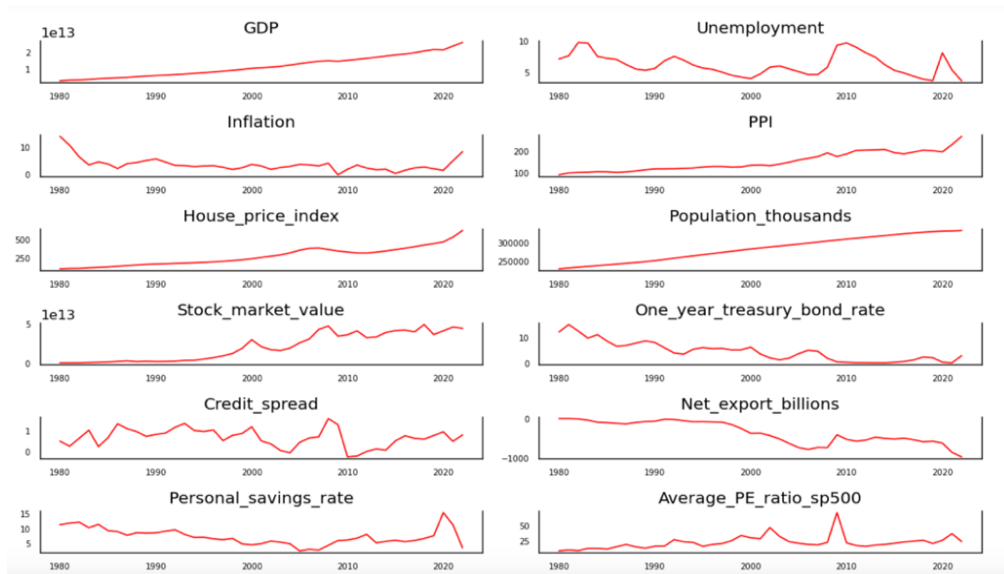


Figure 1: Visualization of data used to train the models.

Aside from very common economic indicators such as unemployment, inflation, PPI, net export, personal savings rate, and treasury bond rate, this research also includes several less conventional indicators.

House price index is included because house is a major source of wealth for common people. A higher house price will increase the wealth of consumers, thus motivate consumption and drive GDP growth. Total stock market value is included for similar reason: stock is also a major source of wealth, especially in the US, where the proportion contributed by stock market in firms' financing process is the highest in the world.

The credit spread is the difference in interest rate between a collection of AAA-rated bonds and the treasury bond rate, reflecting the return of risk in the economy. A higher return on risk will motivate investment and drive GDP growth.

The average PE ratio of SP500 companies reflects the profitability of the average company in the US stock market. Higher profitability of an average company would mean a more prosperous economy and higher GDP growth.

#### 4. Prediction Results

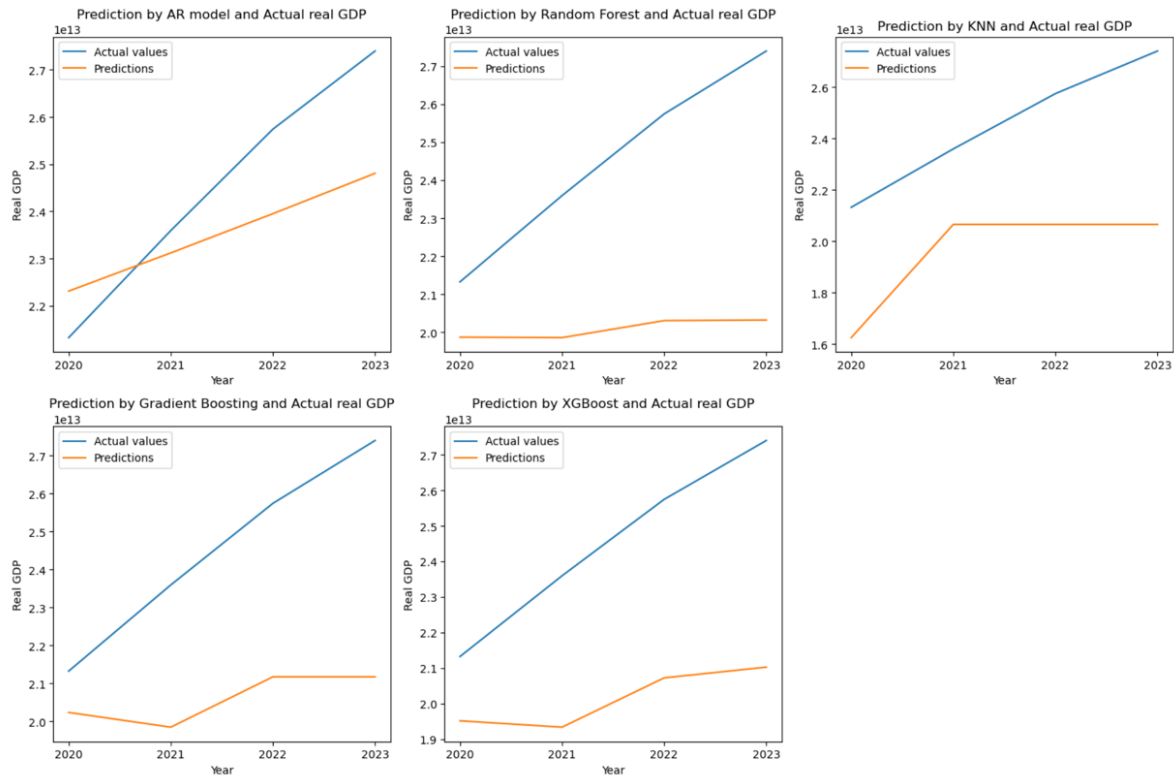


Figure 2: Predicted and Actual real GDP of all models.

The MAEs for the predictions are presented in table 1.

Table 1: MAE of the predictive models.

| Model             | MAE as percentage of average real GDP in the testing data set |
|-------------------|---------------------------------------------------------------|
| AR (1)            | 5.96%                                                         |
| Random Forest     | 18.1%                                                         |
| KNN               | 20.2%                                                         |
| Gradient Boosting | 15.9%                                                         |
| XGBoost           | 17.8%                                                         |

As shown in table 1, the AR (1) model significantly outperforms the other models in predicting US real GDP after covid-19 outbreak. And from the graphs in figure 1, except for the AR model when predicting the real GDP for 2020, all models underestimate the actual value.

The covid-19 is a sudden shock to the US economy, and no economic indicator before 2020 can signal it. The economic stimulus package the US government implemented afterwards is also unforeseeable. Therefore, the models trained using data before 2020 assume there to be no shock and no sudden economic stimulus to the economy, predicting the real GDP to be increasing moderately from 2020 to 2023. However, in 2020, the US government implemented many expanding fiscal and monetary policy including direct payments to all citizens and reducing the interest rate to nearly zero. Such stimulating policies led to an unprecedented high GDP growth rate from 1980. Therefore, the models trained using pre-pandemic data underestimated the real GDP.

But the AR (1) model uses not only pre-pandemic data but also post-pandemic data in its prediction. Its prediction of US real GDP in 2021, 2022, and 2023 is based on the US real GDP in 2020, 2021, and 2022 respectively. Therefore, the high GDP growth rate after 2020 is captured in AR model's prediction, contributing to a lower MAE compared to the other models.

## 5. Limitations

The data used in this research are all annually collected data. With the data collecting frequency of one-year, short term trends are unable to be reflected, especially for a sudden shock like the covid outbreak. The annual GDP does not show the decrease at the beginning of 2020 due to factors such as lockdowns, public concerns, and production chain disruptions. The usage of quarterly or monthly data can better capture and reflect those short-term trends.

This research only evaluated a few predictive models. Traditional models such as vector autoregressive model is similar to the AR model but can incorporate multiple variables from the past years to predict GDP. Machine learning algorithms such as long-short term memory (LSTM) model are more complex and can identify more complex patterns in the training data. This research also attempted to use LSTM but failed to train a credible model with limited error because of the small size of data set. A larger dataset of quarterly or monthly data may help resolve such problem as well.

From the methodology section where the mechanism of the machine learning algorithms used in the research were discussed, we can see that the algorithms used in this research is not precisely forecasting GDP but actually nowcasting it. They take the input of the economic indicators of one year and predict the real GDP of the same year. And it would not be intuitive to forecast one year's GDP using the economic indicators of the previous year as it would be hard to argue the causal relationship in between. A machine learning algorithm specialized for time series data forecasting may be able to resolve this issue.

## 6. Conclusion

The AR (1) model outperforms the random forest, KNN, gradient boosting, and Xgboost algorithms by yielding a lower MAE in the prediction result. The reason behind may be attributed to AR model's ability to utilize data after a sudden shock. Since AR model predict one years' GDP based on the previous year's GDP, after a sudden shock, the effect of the shock will be captured when predicting the GDP of the year after the shock. But the machine learning algorithms evaluated in this research are trained using data in a fixed time frame. They are not able to utilize data in the testing data set and adjust accordingly. By evaluating the performance of the models, this research reveals the importance of a model's ability to handle information of a sudden shock in GDP forecasting and that AR model possess such ability while random forest, KNN, gradient boosting, and Xgboost do not.

## References

- [1] Scott R. Baker, Nicholas Bloom, Steven J. Davis, Stephen J. Terry. (2020) Covid-induced Economic Uncertainty. NBER Working Paper No. 26983, <https://doi.org/10.3386/w26983>
- [2] Raj Chetty, John N. Friedman, Nathaniel Hendren, Michael Stepner, and the Opportunity Insights Team. (2020) The Economic Impacts of COVID-19: Evidence from a New Public Database Built Using Private Sector Data. NBER Working Paper No. 27431, <https://doi.org/10.3386/w27431>
- [3] Giorgio E. Primiceri, Andrea Tambalotti. (2020) Macroeconomic Forecasting in the Time of COVID-19. Manuscript, Northwestern University. Retrieved from <https://api.semanticscholar.org/CorpusID:220323476>
- [4] Sydney C. Ludvigson, Sai Ma, and Serena Ng. (2020) COVID-19 and The Macroeconomic Effects of Costly Disasters. NBER Working Paper No. 26987, <https://doi.org/10.3386/w26987>
- [5] Yutaka Kurihara, Akio Fukushima. (2019) AR model or machine learning for forecasting GDP and consumer price for G7 countries. *Applied Economics and Finance*, 6(3):1, <https://doi.org/10.11114/aef.v6i3.4126>

- [6] Giovanni Maccarrone, Giacomo Morelli, Sara Spadaccini. (2021) *GDP Forecasting: Machine Learning, Linear or Autoregression?* *Frontiers in Artificial Intelligence*, Volume 4, <https://doi.org/10.3389/frai.2021.757864>
- [7] Marta Bańbura, Domenico Giannone, Michele Modugno, Lucrezia Reichlin. (2013) *Chapter 4 - Now-Casting and the Real-Time Data Flow*. *Handbook of Economic Forecasting*, Volume 2, Part A, Pages 195-237, <https://doi.org/10.1016/B978-0-444-53683-9.00004-9>.
- [8] Serena Ng. (2013) *Chapter 14 - Variable Selection in Predictive Regressions*. *Handbook of Economic Forecasting*, Volume 2, Part B, Pages 752-789, <https://doi.org/10.1016/B978-0-444-62731-5.00014-2>.
- [9] Badi H. Baltagi. (2013) *Chapter 18 - Panel Data Forecasting*. *Handbook of Economic Forecasting*, Volume 2, Part B, Pages 995-1024, <https://doi.org/10.1016/B978-0-444-62731-5.00018-X>.
- [10] Marcelle Chauvet, Simon Potter. (2013) *Chapter 3 - Forecasting Output*. *Handbook of Economic Forecasting*, Volume 2, Part A, Pages 141-194, <https://doi.org/10.1016/B978-0-444-53683-9.00003-7>.
- [11] Biau, G., Scornet, E. (2013) *A random forest guided tour*. *TEST* 25, 197–227. <https://doi.org/10.1007/s11749-016-0481-7>
- [12] Natekin Alexey , Knoll Alois. (2013) *Gradient boosting machines, a tutorial*, *Frontiers in Neurorobotics*, Volume 7, <https://doi.org/10.3389/fnbot.2013.00021>
- [13] Chen, Tianqi, Guestrin, Carlos. (2016) *XGBoost: A Scalable Tree Boosting System*, *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794. <https://doi.org/10.1145/2939672.2939785>