INFLATION FORECASTING WITH COMMODITIES

ALVIN WANG / DAYOU LI / TIANJIAN CHE / XIAOYANG CHI

Abstract. Inflation has not been a serious and persistent economic problem in developed country for decades. However, in the wake of the Covid-19 pandemic, fears of inflation have returned. Where is inflation headed, and how can we use what we know to predict where it will go in real time? The causes and effects the variations of inflation are raising peoples' concern. Today we try to use machine learning models in order to find a more accurate measurement method of nowcasting the inflation rates with pricing of commodities.

1. Data collection and processing

The raw historical data that used to feed our machine learning model can be sourced from Kaggle, which includes the daily prices of comodity (Gold, Palladium, Nickel, Brent Oil, Natural Gas and Wheat) from 2000 to March, 2022 and the CPI data of United States.

We focused on data cleaning as part of feature engineering. The removal of missing values (NaNs) might cause the decreasing of model interpretability and reliablity, so we filt each missing value with its next valid value. Besides, for the mixed frequency dataframe, we interpreted X in days, and Y in months, and then took monthly averages on X.

The following formula would be used to calculate year-to-year growth rate:

$$\text{Growth} = \frac{(\text{this year CPI} - \text{last year CPI})}{\text{last year CPI}}$$

FIGURE 1. Inflation

FIGURE 2. Year-to-Year Growth Rate of CPI

2. First attempt: Ordinary Linear Regression

As the naivest and most intuitive method, we firstly applied OLS (ordinary least squares). The establishment of the model was relatively simple: firstly, we split the data set into training data set (before 2017) and testing data set (after 2017), and then ran the LinearRegression model from the sklearn library, which returned the following outcome.

FIGURE 3. Linear Regression

3. Linear regression with regularizations

Adding LASSO or Ridge regularizations could be an effective way to reduce probability of overfitting, the following graphs shows the performanaces of adding LASSO regularization and Ridge regularization respectively.

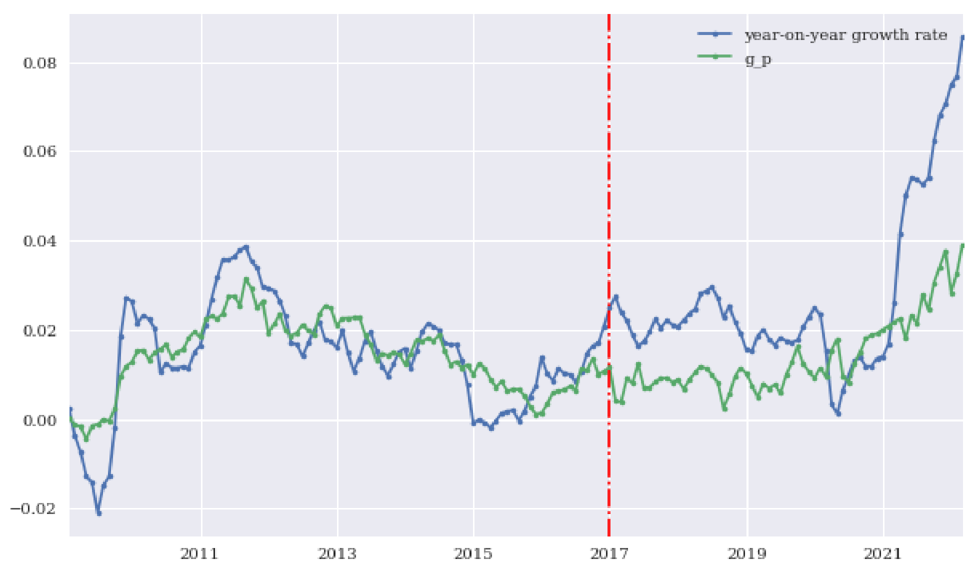

FIGURE 4. With Lasso Regularizations

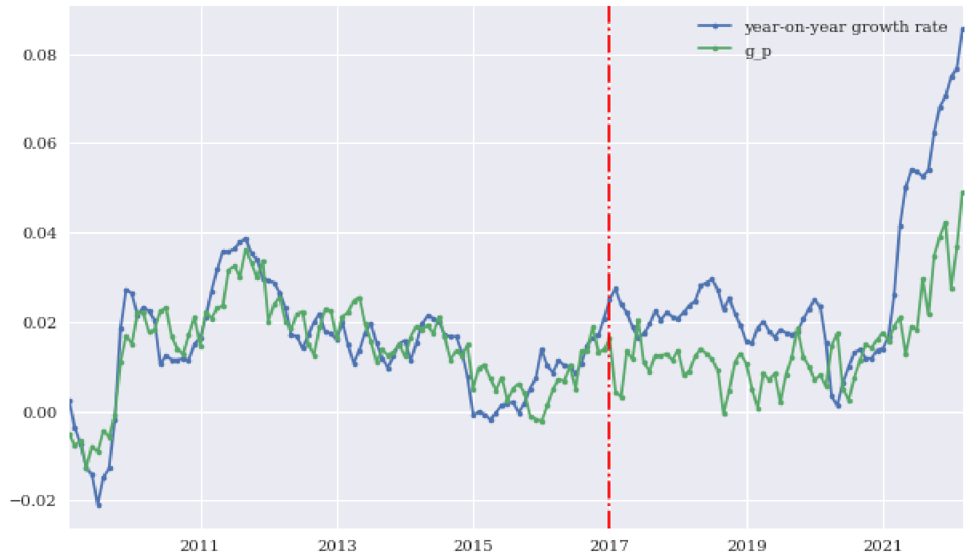

FIGURE 5. With Ridge Regularizations

4. APPLYING OTHER MACHINE LEARNING MODELS

In this section, we tried to apply different machine learning models or parameter modifications as introduced in lectures, including LSTM, Multilayer Perceptron, Random Forest, Decision Tree and XGBoost.

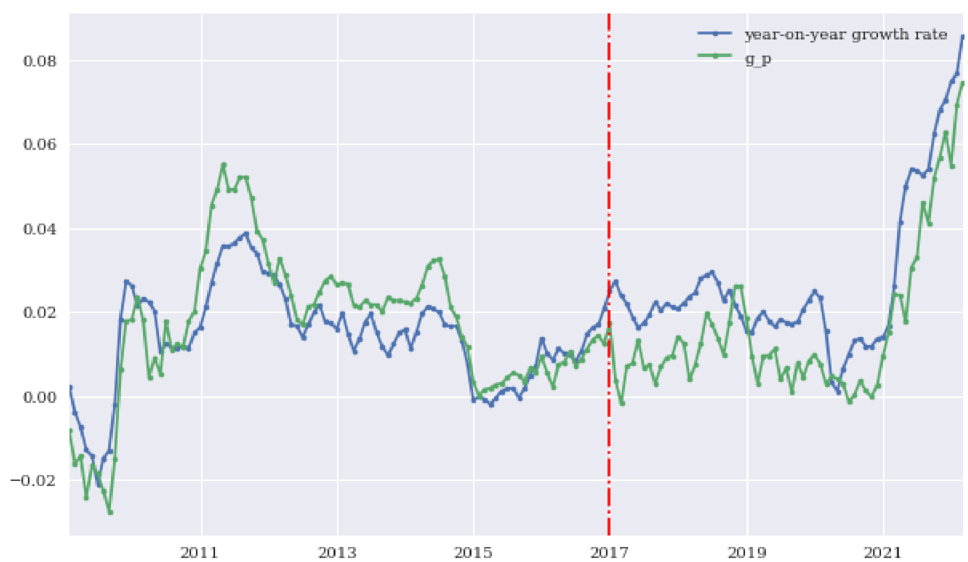

FIGURE 6. LSTM

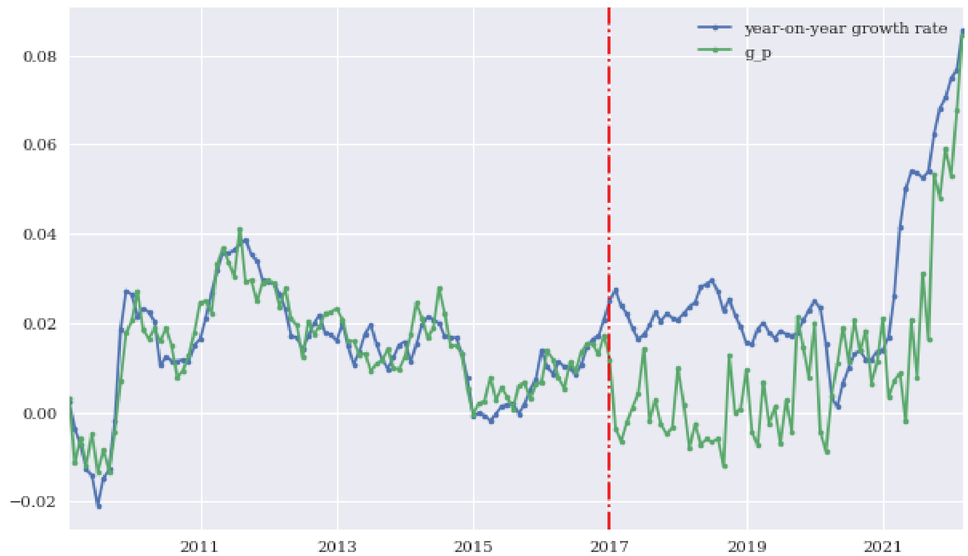

FIGURE 7. Multilayer Perceptron with 10 layers

FIGURE 8. Random Forest

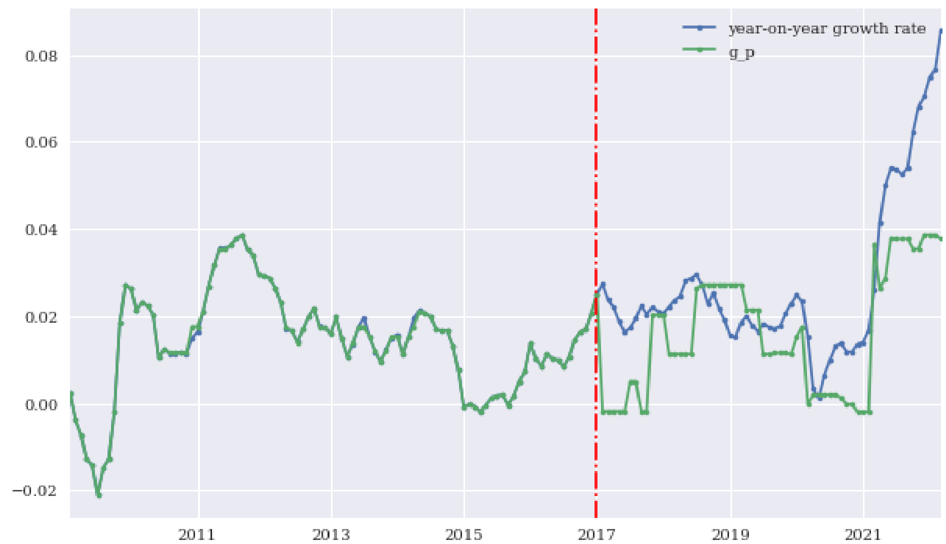

FIGURE 9. Decision Tree

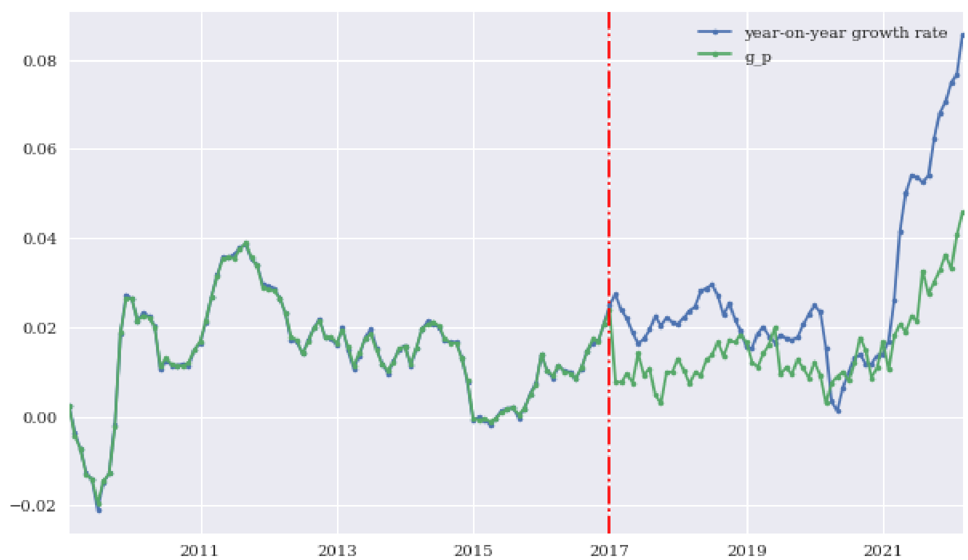

FIGURE 10. XGBoost

5. Forcasting Summary

As our model summary, the following table generates MSEs for each machine learning model we chose.

	Training MSE	Testing MSE
OLS	$2.9104948516751224 \times 10^{-5}$	0.0003298666400212202
LassoCV	$5.0932319589503475 \times 10^{-5}$	0.00035088866067721177
RidgeCV	$3.68902460248577 \times 10^{-5}$	0.00033034868436919413
LSTM	$6.995119679684246 \times 10^{-5}$	0.00017586422637605402
MLP	$2.383843843916805 \times 10^{-5}$	0.0004979056514379927
Random Forest	$3.929795329486218 \times 10^{-6}$	0.00029031614677209805
Decision Tree	$1.8067843114297657 \times 10^{-7}$	0.00028472309805151636
XGB	$2.0120924594331378 \times 10^{-7}$	0.0002871991363157281

Table 1. Model Summary

6. Nowcasting summary

The following graphs generate the nowcasting outcomes and LSTM result.

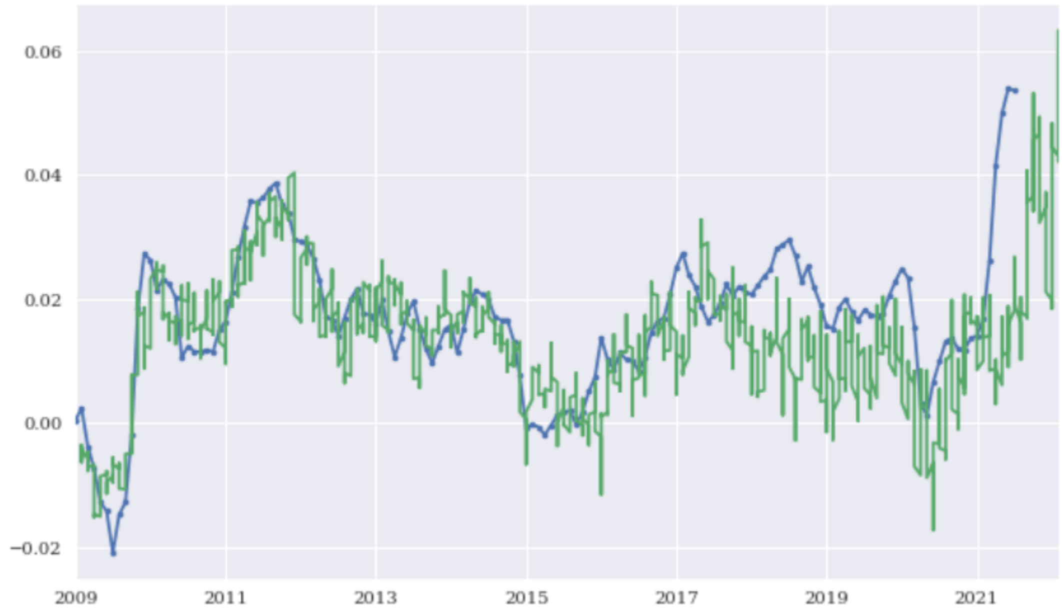

FIGURE 11. SMA outcome

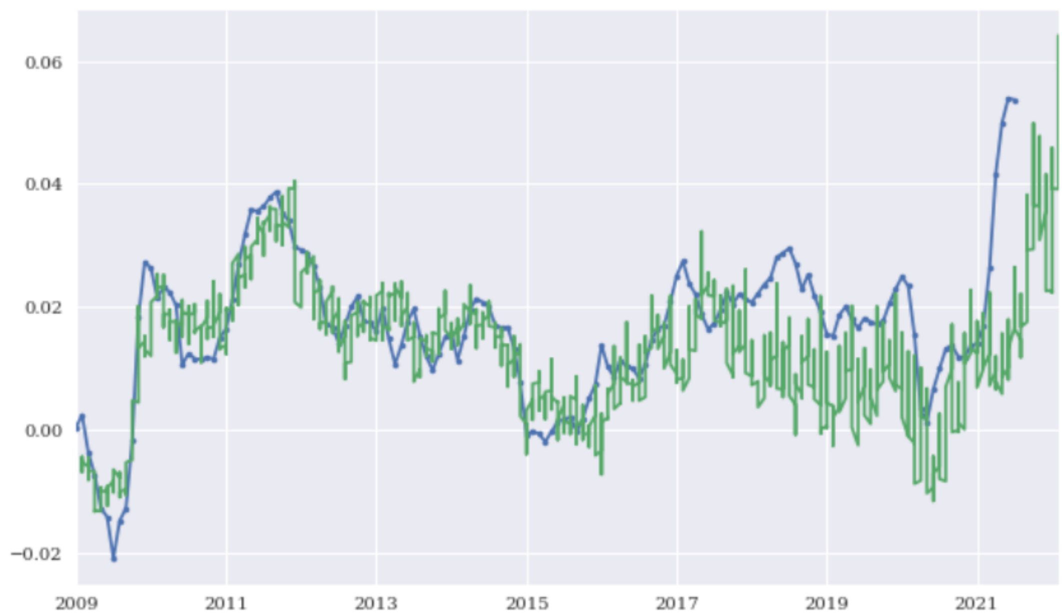

FIGURE 12. EMA outcome

FIGURE 13. LSTM result