Forecasting analysis S&P 500 Retrun (1976 – 2016)

Zhihao Liu

1. Definition

Project overview

Based on economic theory that stock index has strong link between many macroeconomic variables and its historical variation. During this project, we create a dataset contains S&P 500 Returns monthly from 1976 to 2016 and 8 predictive variables. Majority of all variables were found on the St. Louis Federal Reserve website. ISM PMI index was found on our Reuters terminal. There are 492 observations in the dataset and all variables are numeric. The goal of this project is to find the best algorithm that outperform than others in forecasting.

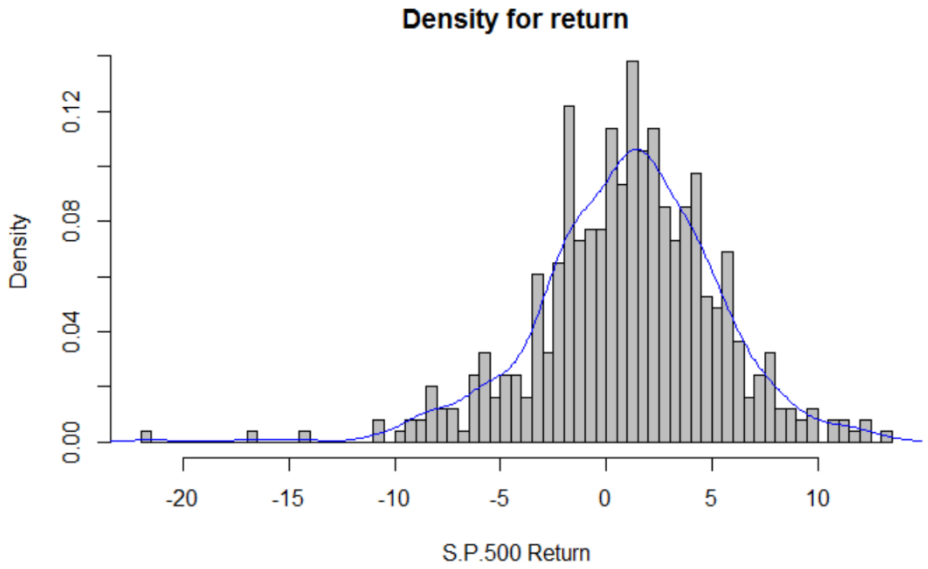
Algorithm and metric

forecasting the S&P 500 Return is a time series and regression problem. I choose ARIMA model as the benchmark algorithm, because ARIMA model can combine time series index and predictive variables to do analysis. And it can give us the number of lagged period that we should consider. Then, we should add the lagged indicators for other models. I also try KNN, General Additive Model with Spline, Random Forest and Boosting.

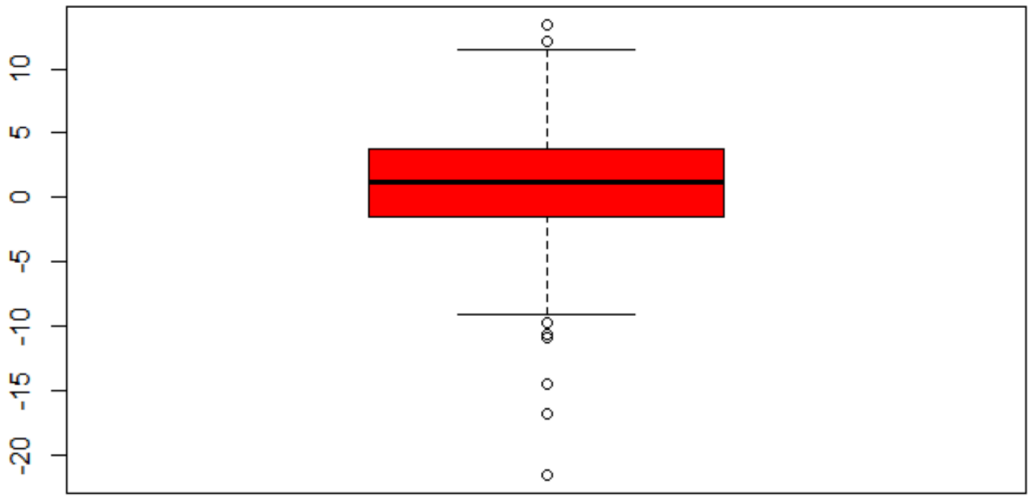
In this project, the metric evaluation is Mean Square Error (MSE).

2. Data Exploration

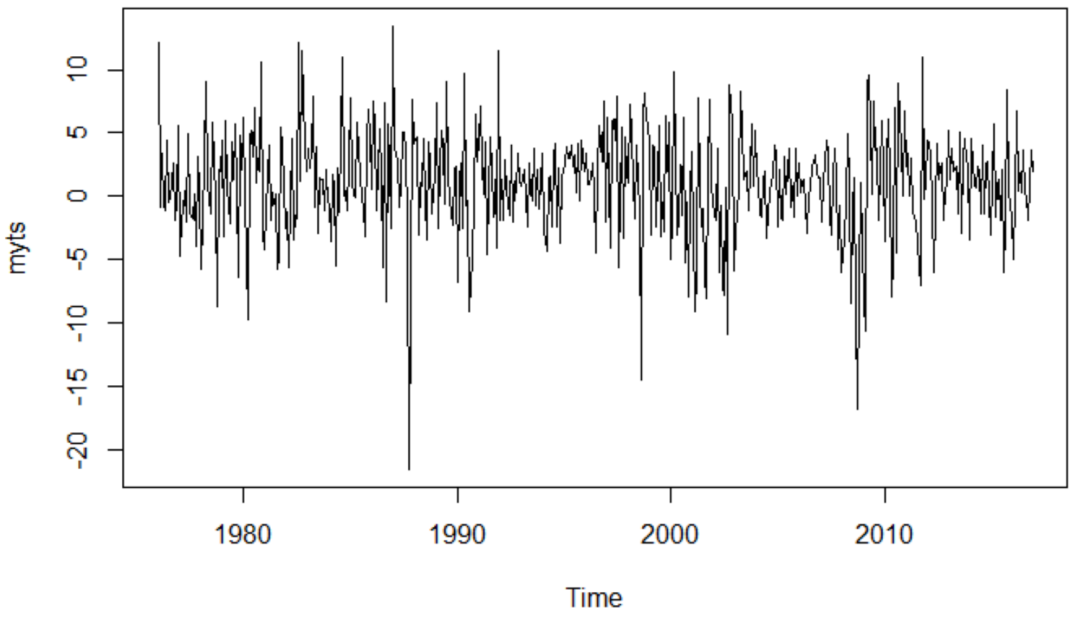
Firstly, I try to find the density distribution of S&P 500 Returns. The Histogram is shown below. We can see that most observations range from -10% to 10%. And the density distribution is seem to follow normal distribution.



Then we can see the boxplot below, the mean of S&P Returns is around 0 and the range from the first quartile to the third quartile is (-4, 4). And there are 5 outliers smaller than -10% and 2 outliers larger than 10%.



And below is Variation Image, we can see the change of S&P 500 Returns from 1976 to 2016 is roughly flat around 0, and there is no significant trend and seasonality.



3. Analysis

Split data

Firstly, I try to split data in to training and testing. Because this is a time series dataset, I use the data from 1976 to 2000 as train dataset and data from 2001 to 2016 as test dataset.

ARIMA Model

The acronym ARIMA stands for Auto-Regressive Integrated Moving Average. Lags of the stationarized series in the forecasting equation are called "autoregressive" terms, lags of the forecast errors are called "moving average" terms, and a time series which needs to be differenced to be made stationary is said to be an "integrated" version of a stationary series. ARIMA model the most general class of models for forecasting a time series dataset.
Firstly, I checked if the data is stationary.

```
p-value smaller than printed p-value
        Augmented Dickey-Fuller Test

data:  myts
Dickey-Fuller = -7.3993, Lag order = 7, p-value = 0.01
alternative hypothesis: stationary
```

Based on the result of Augmented Dickey-Fuller Test, we can see that the P-value is less than 0.05, which means we can accept the alternative hypothesis that the data follows stationary.

Then, let's see the summary of ARIMA model. We can see auto.arima function chooses lagged value equal 1 which means the t-1 S&P 500 Return has influence for forecasting. And based on the result of coefficients we can see variables **ar1, ISM, DGS10, ALTSALES, HSN1F and CSUSHINSA** have negative influence on the change of stock index and **FEFUNDS, CPIAUCSL and UMCSENT** have positive influence on it.

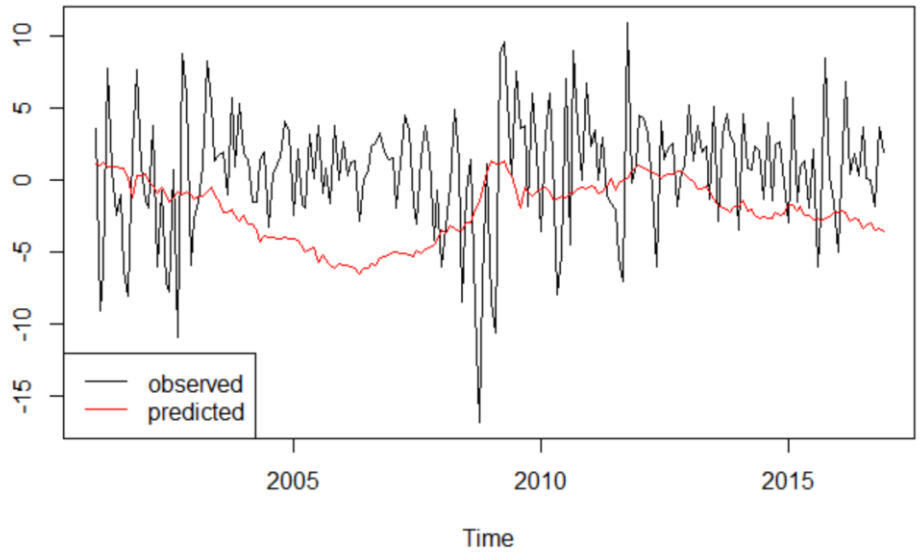
```
Series: train_y
ARIMA(1,0,0) with non-zero mean

Coefficients:
          ar1  intercept      ISM  FEDFUNDS    DGS10  CPIAUCSL  UMCSENT  ALTSALES
      -0.0607    10.7191  -0.1065    0.0558  -0.3672    0.0431   0.0382   -0.2431
s.e.   0.0593     3.8271   0.0471    0.1632   0.2821    0.0420   0.0348    0.2381
        HSN1F  CSUSHPINSA
      -0.0001     -0.0944
s.e.   0.0036      0.0730

sigma^2 estimated as 18.16:  log likelihood=-855.5
AIC=1732.99   AICc=1733.91   BIC=1773.73

Training set error measures:
                     ME     RMSE      MAE      MPE     MAPE      MASE         ACF1
Training set 0.001993927 4.190031 3.189083 91.97087 160.9453 0.6915823 -0.00323567
```

And the histogram below shows the comparison between the observed stock index from 2001 to 2016 and the predicted value of ARIMA model based on testing data. We can see that the performance is not good, in some periods the trends are totally different.

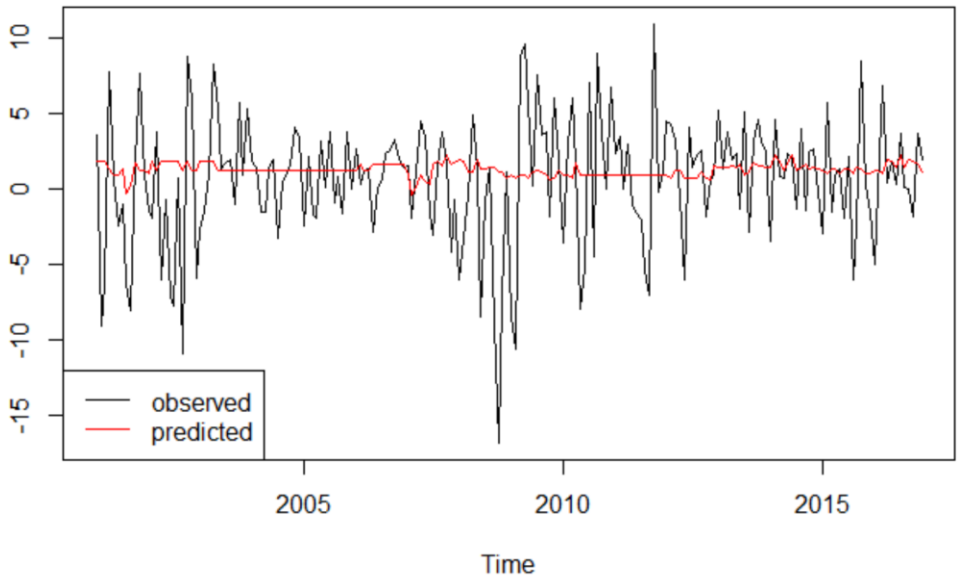


Finally, I find the test MSE of ARIMA model is **29.40489**

K-Nearest Neighbor Method

Because in ARIMA model, the t-1 value is considered. So I add this as a new variable in KNN method. And after cross-validation, k=20 is chosen.

The histogram below shows the comparison between the observed stock index from 2001 to 2016 and the predicted value of KNN model based on testing data. we can see it is better than ARIMA, but it is too flat, some significant changes cannot be reflected.



the test MSE of KNN model is **18.64799.**

General Additive Model (GAM) and Spline combination

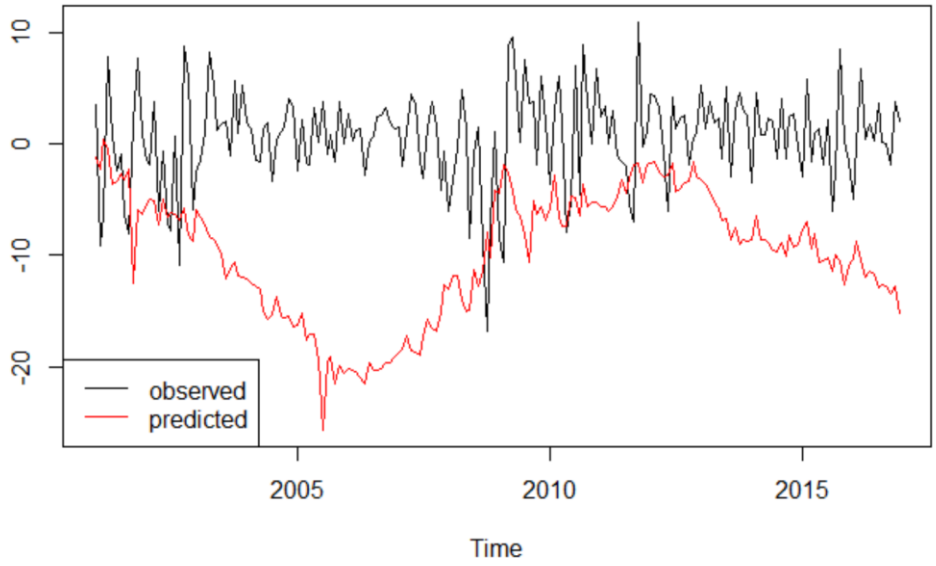
In this dataset, we know that the relationship between the predictive variables and response is non-linear, so I consider to use non-linearity analysis method for multiple independent variables. GAM with spline of different variables is my choice.
Firstly, I use smooth.spline Function to find the degree of freedom for every predictive variables.

```
[1] 2.005546 2.829615 3.541058 2.014927 2.016908 4.913939 2.014742 2.574179 4.325691
```

Then, I built the GAM model based on training data,

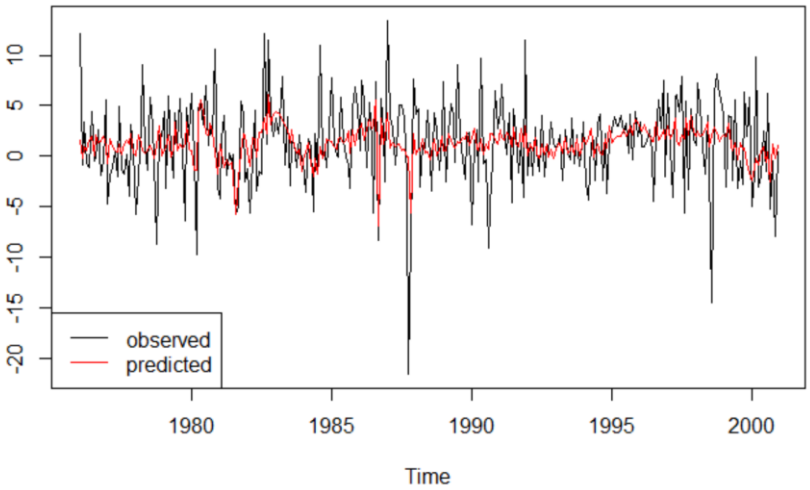
```
gam(y~s(x.ISM,2.005546)+s(x.FEDFUNDS,2.829615)+s(x.DGS10,3.541058)+s(x.CPIAUCSL,2.014927)+
s(x.UMCSENT,2.016908)+s(x.ALTSALES,4.913939)+s(x.HSN1F,2.014742)+s(x.CSUSHPINSA,2.574179)+
s(x.t.1.return,4.325691),data=train.data)
```

The histogram below shows the comparison between the observed stock index from 2001 to 2016 and the predicted value of the model combine GAM and splines based on testing data. we can see that the performance is really bad, the predicted value is very different with observed value.



the test MSE of GAM model is **157.8719.** This error is really big, I think maybe there exist overfitting problem for this model, so I want to check the train MSE and histogram for train data.

The histogram below shows the comparison between the observed stock index from 2001 to 2016 and the predicted value of the model combine GAM and splines based on training data. the histogram shows that the model performs really good, the predicted value is simulated to the variation of observed values.

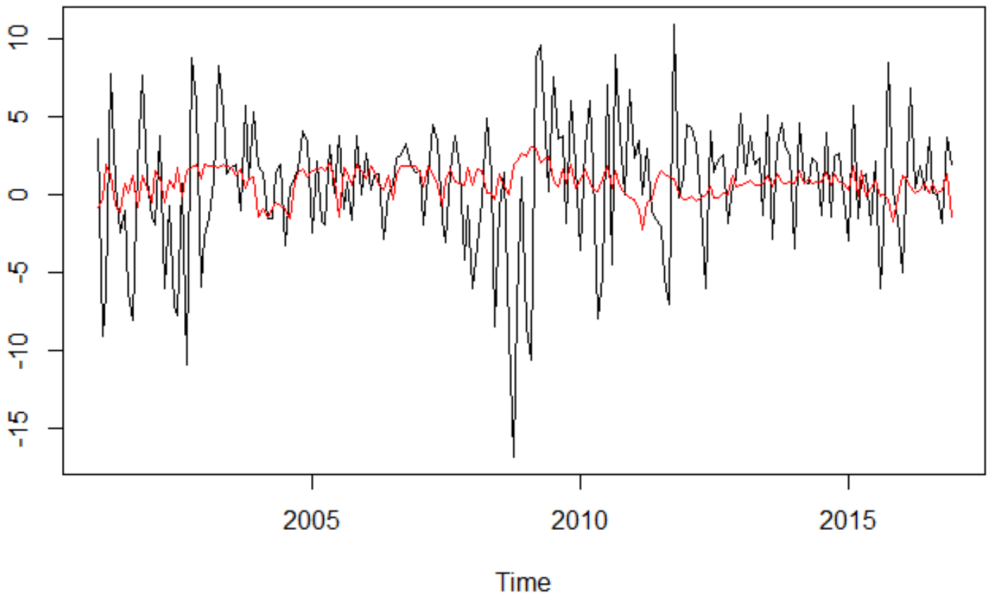


the train MSE of GAM model is **14.89083.** We can see that it is much smaller than test MSE, which means this model is overfitting. The reason for this problem I think may be the dataset is not large enough.

Random Forest

Random forest is a good method in both regression and classification. During this model, I set number of trees equal to 500, number of variables considered at each split equal to p/3=3, terminal node size equal to 5.

The histogram below shows the comparison between the observed stock index from 2001 to 2016 and the predicted value of the random Forest based on testing data. we can see the performance is good, predicted values can show the roughly trend of observed values.

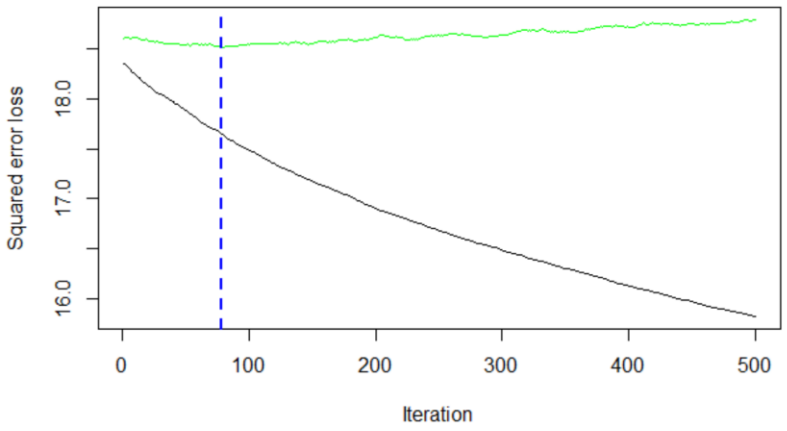


the test MSE of Random Forest model is **18.53256.**

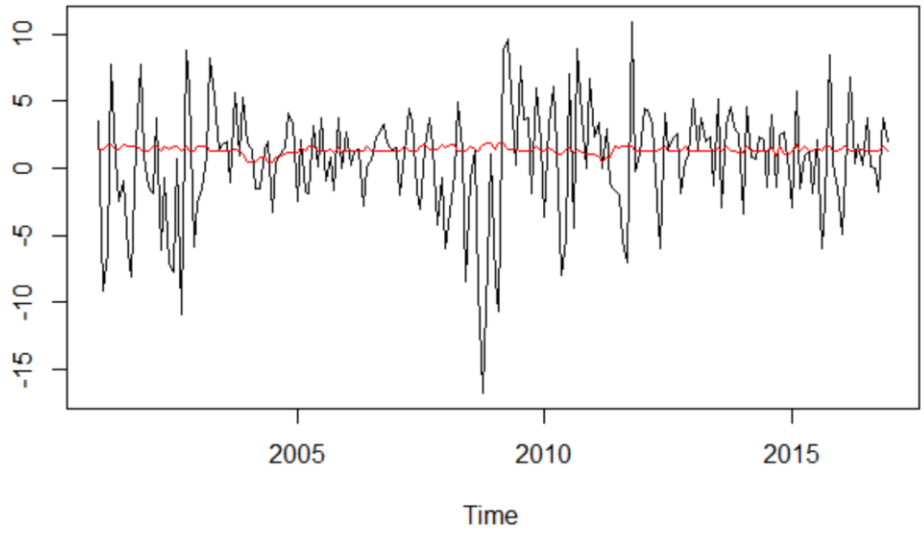
Boosting

This dataset is a time series dataset, so I think maybe boosting can increase the forecasting accuracy. Because in Boosting, each tree is grown using information from previously grown trees.

Firstly, I use cross validation to choose the number of trees that can be used in this dataset. The histogram below shows that number of trees around 80 can lead the smallest MSE.



Then, I build the Boosting model, the histogram below shows the result. The performance is similar with KNN model, the change of predicted value is too flatted.



And the test MSE of Boosting is **18.83883.**

4. Conclusion

	ARIMA	KNN	GAM with Spline	Random Forest	Boosting
MSE(test)	**29.40489**	**18.64799**	**157.8719**	**18.53256**	**18.83883**

According to the result, we can see that Random Forest has the best performance. And KNN and Boosting have the similar performance. GAM with Splines has overfitting problem.

5. Future work

Based on the models, I think I should try ensemble method, make a combination of the top three performance models. Because different models can explain different features of the datasets. Combination may increase accuracy.

On the other hand, because the response of this dataset has no trend and seasonality, it is flat spread from 1976 to 2016, I think maybe creating a binary outcome “the stock index increases or decreases” to do analysis based on these predictive variables is more meaningful.