BDS-Assignment2

Author: Jingyi Wu(andrew id: jingyiw2)

Used Libraries:

Numpy

Pandas

Matplotlib

Seaborn

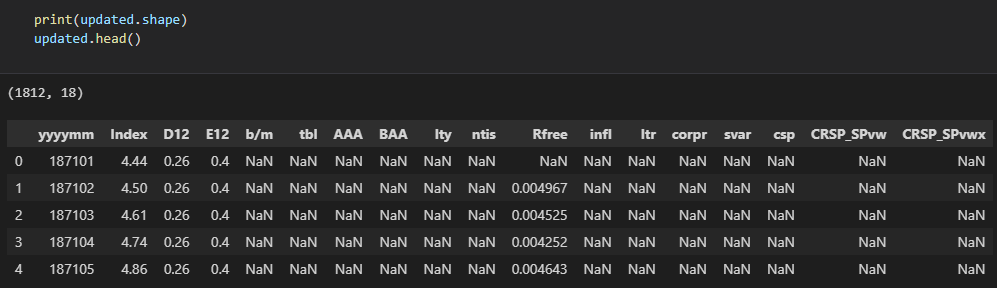
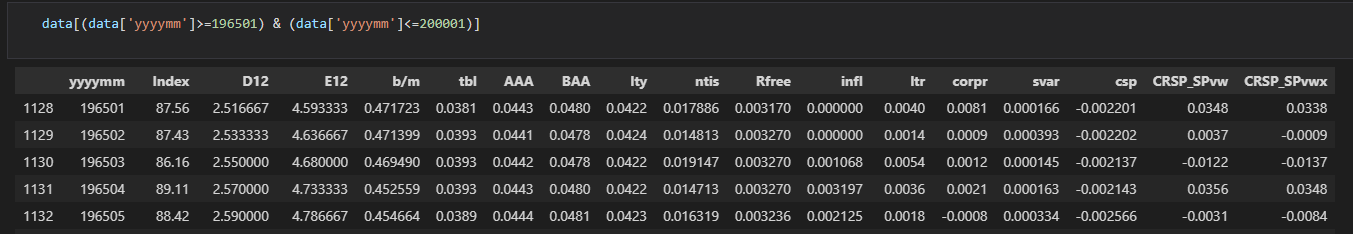
xgboost.sklearn

sklearn

plotly

1. Load the original and updated datasets as “data” and “updated” using pandas.

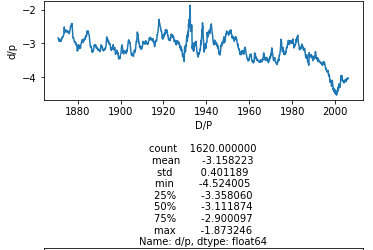
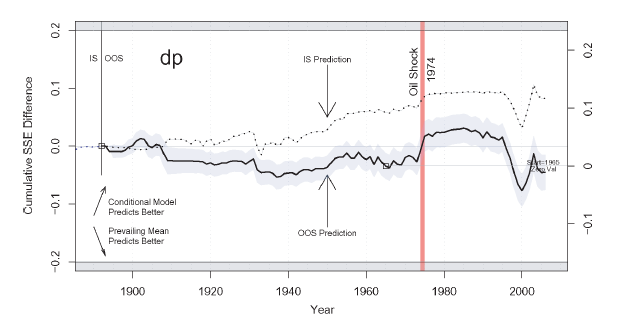
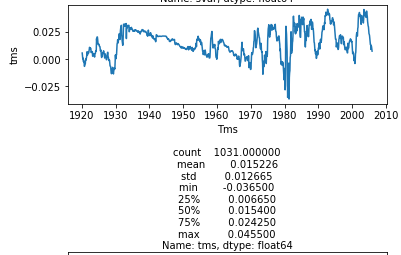
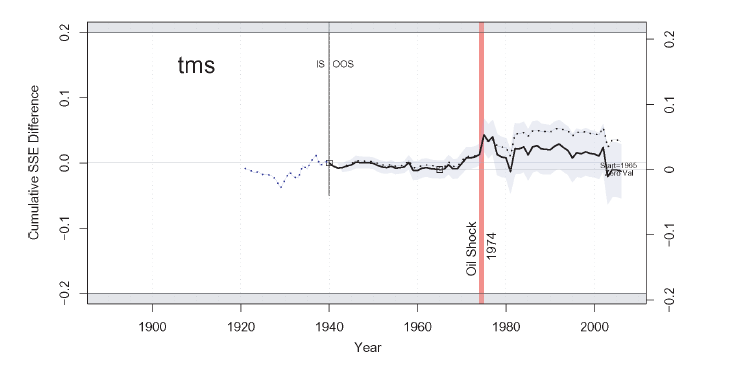
The loading results are as below.

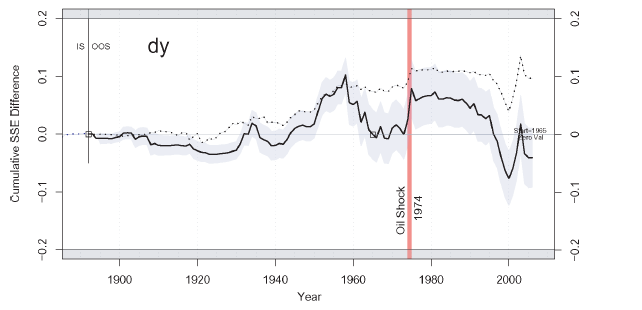


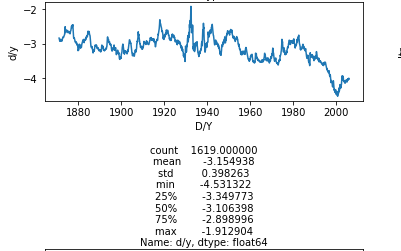
1. Create the predictive variables in the same way as the paper, and generate a new data frame containing these features called ”df”.

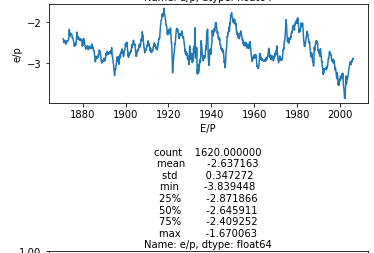
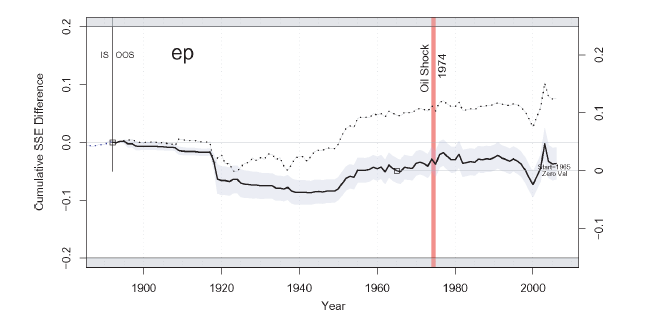
Plot each variable’s monthly data and check with the annual performance of IS insignificant predictors in the paper. Here we use parts of the features to make comparison.

For tms, we can see an increasing trend starting from 1970 and a sharp drop in 1980. For annual data, the line is similar and more smooth.



For dividend-price ratio, similar drop in 2000 and increase in 1974 for actual values and cumulative squared prediction errors. 





For dy and ep, we can find similar pattern, especially after the key event in 1974 happened. In summary, the variables are correctly interpreted.

1. Use rolling multiple linear regression for three time periods data listed and calculate key metrics.

For three time periods data, we generate three data frames, df1, df2 and df3.

To implement rolling regression, here we wrote a function called “rolling\_regression” and for each iteration, we build model and generate predicted fi and mean\_y. After all iteration, we get three arrays, true y values, mean y values, and preficted y values fi.

For MSEa, . For MSEn, . We get R2, and get MAE using sklearn metric function. The results are as below.

Rolling Linear Regression Results

|  |  |  |  |
| --- | --- | --- | --- |
|  | R2 | RMSE | MAE |
| 1965.01-2008.12 | -0.402222 | 0.0513366 | 0.037625 |
| 1976.01-2008.12 | -0.587618 | 0.0542258 | 0.038781 |
| 2000.01-2008.12 | -2.937812 | 0.0842482 | 0.054425 |

Note that here the R2 value are negative, indicating the model does not have predictive ability at all.

1. To improve the performance, since this is time-series financial data and multi-collinearity between variables exists, we use Lasso regression to see if it generates a better prediction.

Rolling Lasso Regression Results

|  |  |  |  |
| --- | --- | --- | --- |
|  | R2 | RMSE | MAE |
| 1965.01-2008.12 | -0.0160496 | 0.0436995 | 0.033077 |
| 1976.01-2008.12 | -0.0398146 | 0.0438845 | 0.032872 |
| 2000.01-2008.12 | -0.1570206 | 0.0456671 | 0.034215 |

By changing into lasso regression, we can see that the performance improved and error decreased. Lasso regression proves to be a better fit here.

Here I also tried xgb regressor, and the performance is as below.

Rolling XGB Regression Results

|  |  |  |  |
| --- | --- | --- | --- |
|  | R2 | RMSE | MAE |
| 1965.01-2008.12 | -0.8392916 | 0.0587955 | 0.039552 |
| 1976.01-2008.12 | -0.9246126 | 0.0597042 | 0.040805 |
| 2000.01-2008.12 | -3.4790942 | 0.0898521 | 0.047352 |

Before applying the model, I assumed that tree model will not work well in this dataset and the truth is as what I expected. Since tree models are quite sensitive to the number of instances and here we use rolling method which means for some iterations, the train set is quite small and the model has great errors, the XGB performs unsatisfying.