# MACHINE LEARNING CHALLENGE

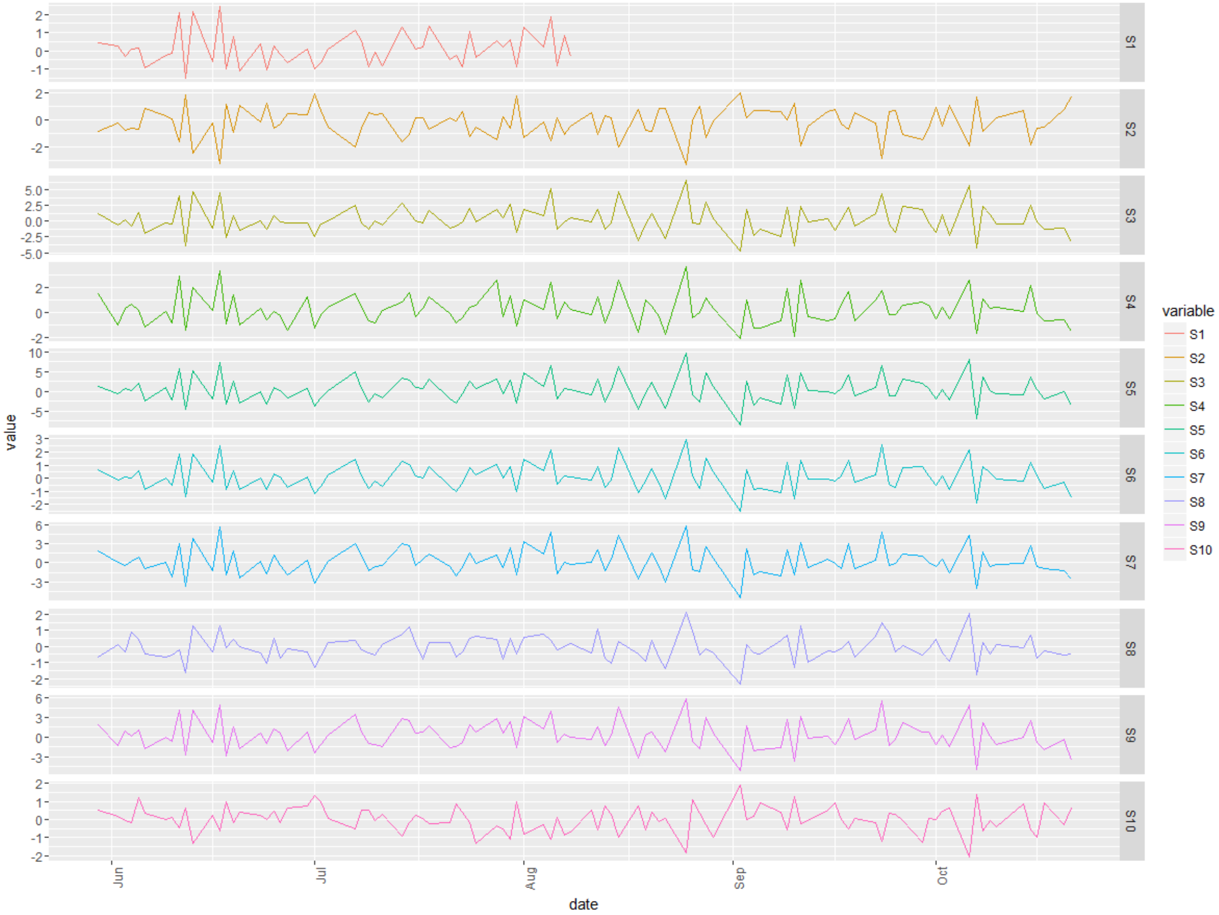
Submitted by: George Wang, August 17, 2016

I explored both linear time series regressions and gradient boosted decision tree method. I decided to choose the tree-based method because of its potential for better out-of-sample performance. The current model is probably overfit, as the out-of-sample performance is much lower than in-sample, but with additional data (both data points and number of features), the decision tree model could potentially be very rewarding.

See below for a brief discussion of the major components of the model development process, including data exploration, feature engineering, variable selection, time series regression, gradient boosted decision trees, performance testing, cross validation, and final output.

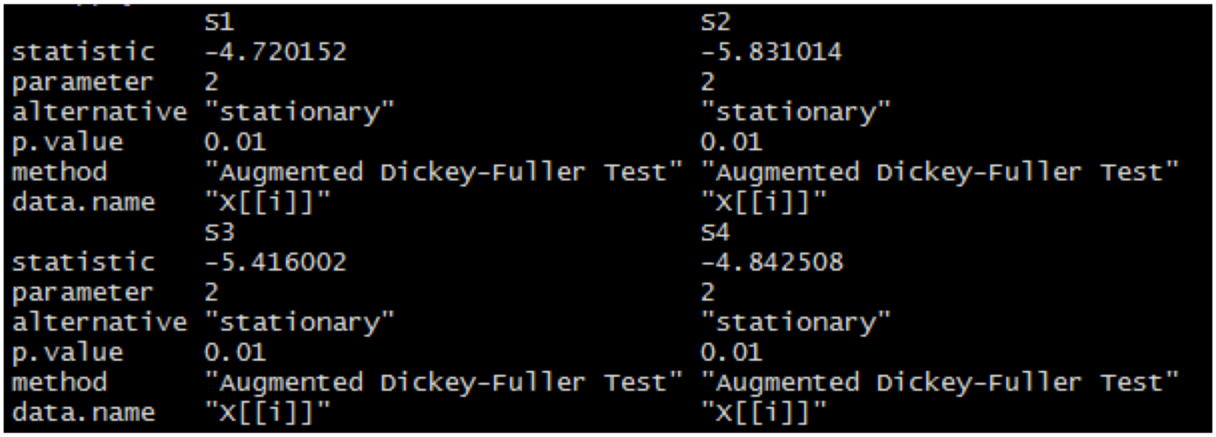
## DATA

First we plot the 10 time-series variables, S1 through S10, to gain an initial understanding of the data.



## FEATURE ENGINEERING

Given that S2 through S10 are traded in the Japanese stock market, which close earlier than its US counterpart, we would expect the variables themselves to have some predictive power in explaining S1. Additionally, we add the first and second lag of each S2 through S10 as potential candidates into our model for comprehensiveness. A stationarity test of lags up to 2 on each variable was performed to avoid spurious regressions. All 10 variables rejected the null hypothesis at the 5% confidence level, and hence are trend stationary.



```
            S1                          S2
statistic   -4.720152                   -5.831014
parameter   2                           2
alternative "stationary"                "stationary"
p.value     0.01                        0.01
method      "Augmented Dickey-Fuller Test" "Augmented Dickey-Fuller Test"
data.name   "x[[i]]"                    "x[[i]]"
            S3                          S4
statistic   -5.416002                   -4.842508
parameter   2                           2
alternative "stationary"                "stationary"
p.value     0.01                        0.01
method      "Augmented Dickey-Fuller Test" "Augmented Dickey-Fuller Test"
data.name   "x[[i]]"                    "x[[i]]"
```

## VARIABLE SELECTION

We explore a few methods of variable selection: Granger causality, Best subset, and Forward stepwise regression.

### *Granger Causality*

One can state that X granger causes Y if the coefficients on X and lags of X are jointly significant in the following regression:

$$y_t = \alpha + \beta_1 y_{t-1} + \beta_2 y_{t-2} + \cdots + \beta_p y_{t-p} + \gamma_1 X_t + \gamma_2 X_{t-1} + \cdots + \gamma_{p+1} X_{t-p} + \epsilon_t$$

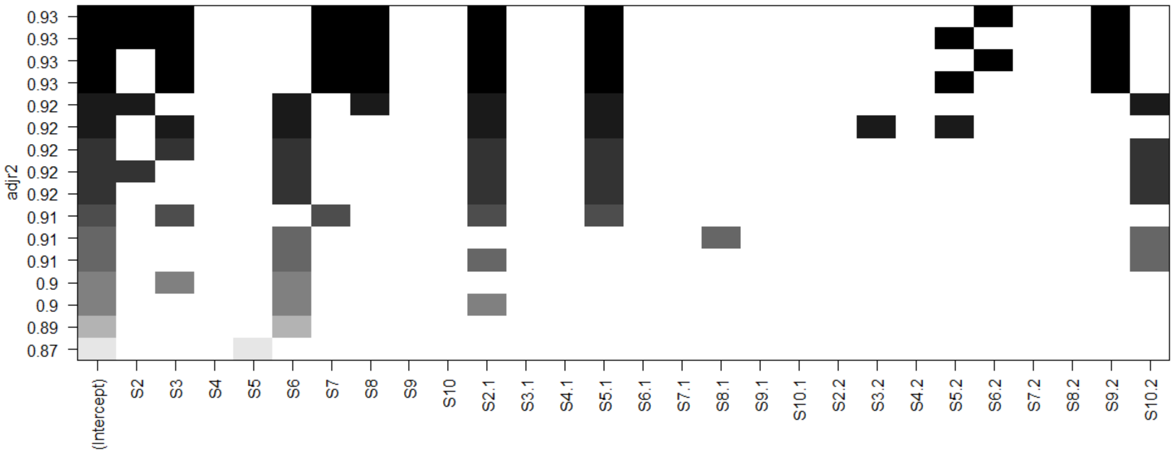
where p is selected based on the largest statistically significant lag of $y$. If a F-test performed on $\gamma_1 = \gamma_2 = \cdots = \gamma_{p+1} = 0$ is significant at some confidence level, then X granger causes Y.

The results indicate that none of the explanatory variables S2 to S10 are significant at the 5% level. However, S2 and S6 demonstrate relative significant compared to other variables.

### *Best Subset*

The best subset method computes all valid regressions as combinations of the independent variables. Hence in a model with N parameters, there are a total of 2^N number of models to check (property of the sum of binomial coefficients).

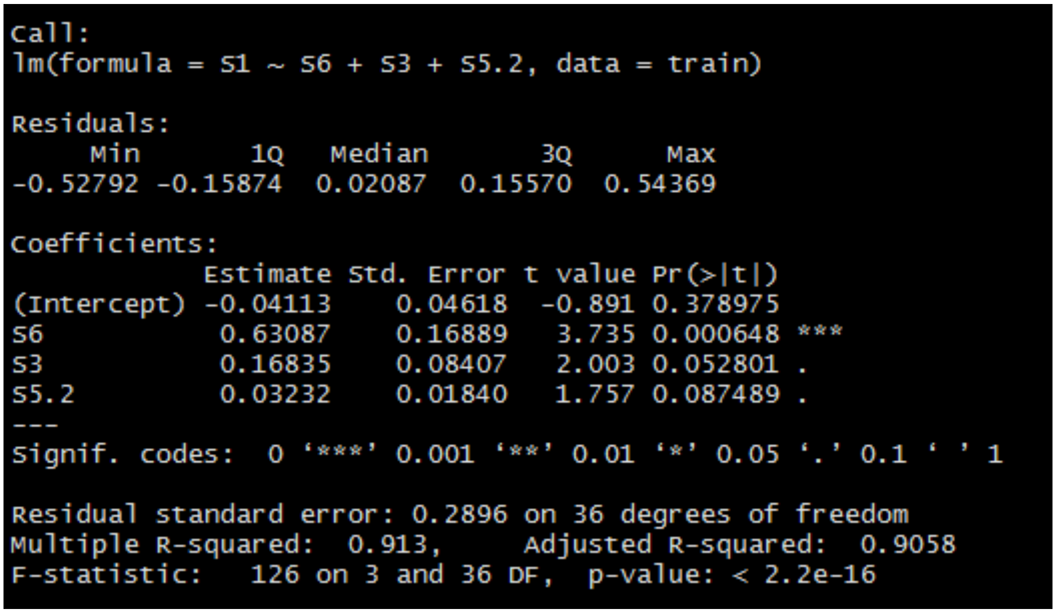
The following figure shows the adjusted R-squared metric corresponding to various models with the chosen variables (in black).



For example, the bottom regression, with just an intercept term and variable S5, has a regression with adjusted R-squared of 0.87. Based on this approach, S2, S3, S7, S8, $S2_{t-1}$, and $S5_{t-1}$ are important factors in prediction S1.

### *Forward Stepwise Selection*

Forward stepwise selection begins with a model that only includes the intercept term. Then during each iteration, the model picks one variable that has the largest improvement on the model, as measured by BIC. The model chosen includes the following variables: $S6, S3, S5_{t-2}$.



```
Call:
lm(formula = S1 ~ S6 + S3 + S5.2, data = train)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.52792 -0.15874  0.02087  0.15570  0.54369 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -0.04113    0.04618  -0.891 0.378975    
S6           0.63087    0.16889   3.735 0.000648 ***
S3           0.16835    0.08407   2.003 0.052801 .  
S5.2         0.03232    0.01840   1.757 0.087489 .  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2896 on 36 degrees of freedom
Multiple R-squared:  0.913,	Adjusted R-squared:  0.9058 
F-statistic:   126 on 3 and 36 DF,  p-value: < 2.2e-16
```

## TIME SERIES REGRESSION

We first explore a linear regression approach, with variables selected using the forward stepwise selection methodology.

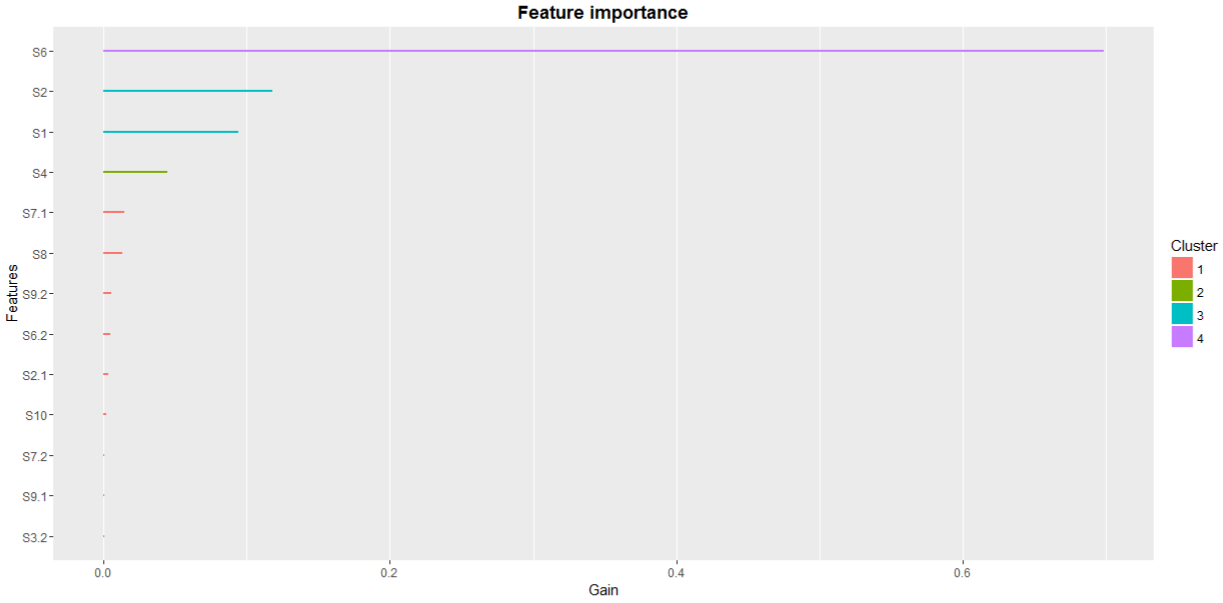
$$E[Y|X] = \alpha + \beta_1 * S6_t + \beta_2 * S3_t + \beta_3 * S5_{t-2}$$

Note that the parameters $\alpha$ and $\beta$ are optimized to minimized sum of squared errors using the OLS method, which is equivalent to the MLE method when the errors are iid Gaussian.

## GRADIENT BOOSTED DECISION TREES

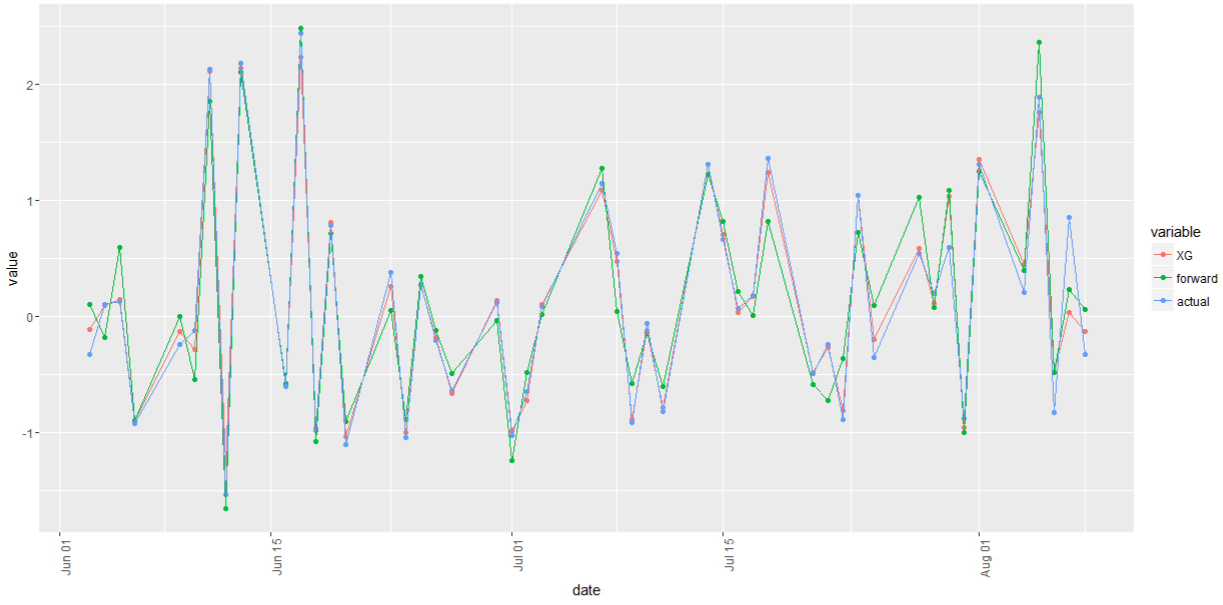
We then explore more modern machine learning techniques, such as the gradient boosted decision tree. A boosted tree is an iterative process that fits additional models to the error of the previous model. Hence it could achieve accuracies greater than linear regression. However, we should be aware of the bias variance tradeoff so as to not overfit.

The XG Boost tree identifies S6, S2, and S1 as the most important features.



## PERFORMANCE TESTING

A visual of the in sample and out of sample performance, using a 40/10 training/testing data spilt.



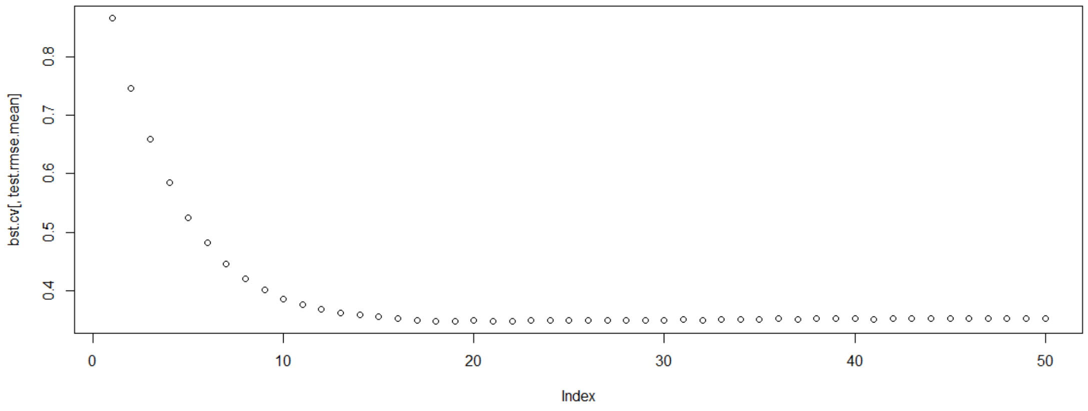
The following table displays the Root Mean Squared Error (RMSE) of the two models for both in sample and out of sample.

| | In Sample | Out Of Sample |
|---|---|---|
| XG Boost | 8.6% | 37.2% |
| Forward | 27.5% | 38.3% |

We observe overfitting in XG Boost, as the out of sample performance is drastically lower than in sample. However, it still out performs the OLS in this set of training/testing data split.

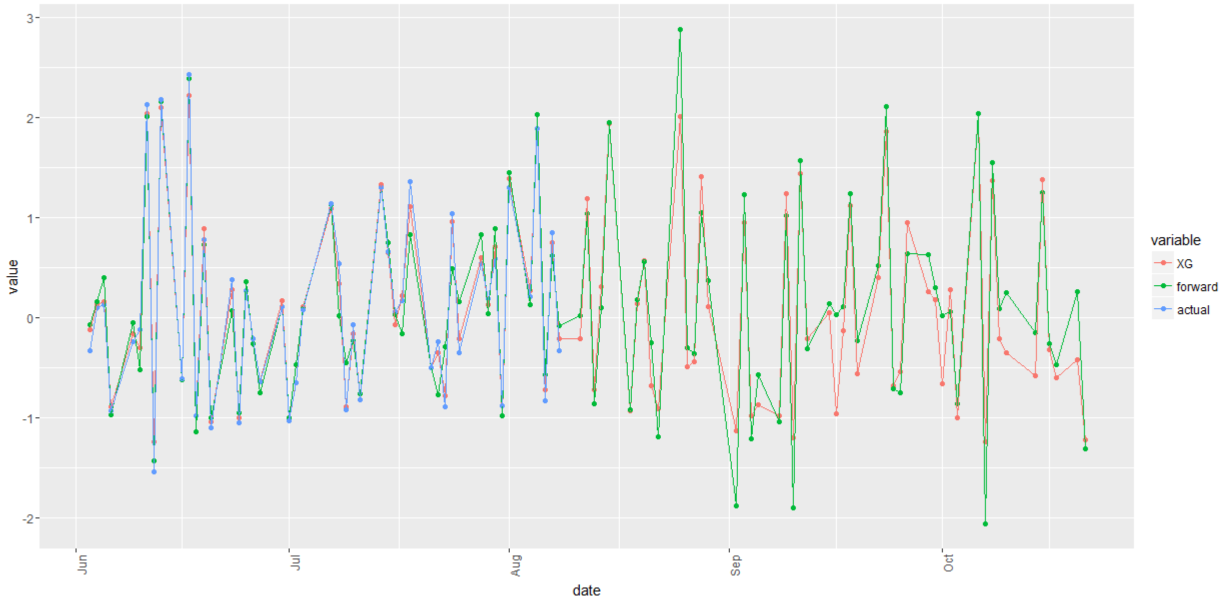
## CROSS VALIDATION / PARAMETER CALIBRATION

To improve the performance of XG Boost, we iterate over number of boost trees, maximum depth of trees, and learning rate to find the optimal parameters that minimizes out of sample RMSE



## OUTPUT

Finally, we use the optimized hyper-parameters and rerun the two models over the entire set of sample data. Below is a visual of XG Boosted tree vs. Forward stepwise selection linear regression.



The in-sample RMSE is displayed below.

| | In Sample |
|---|---|
| XG Boost | 10.5% |
| Forward | 29.5% |