Predicting the NASDAQ Index

April 26, 2021

15.034 Section C/D Group 8 Project 2

James Elgin

Keith Fleming

Shaun Gan

Shu Ge

Kiran Gite

Tico Han

Ke Qiu

Introduction

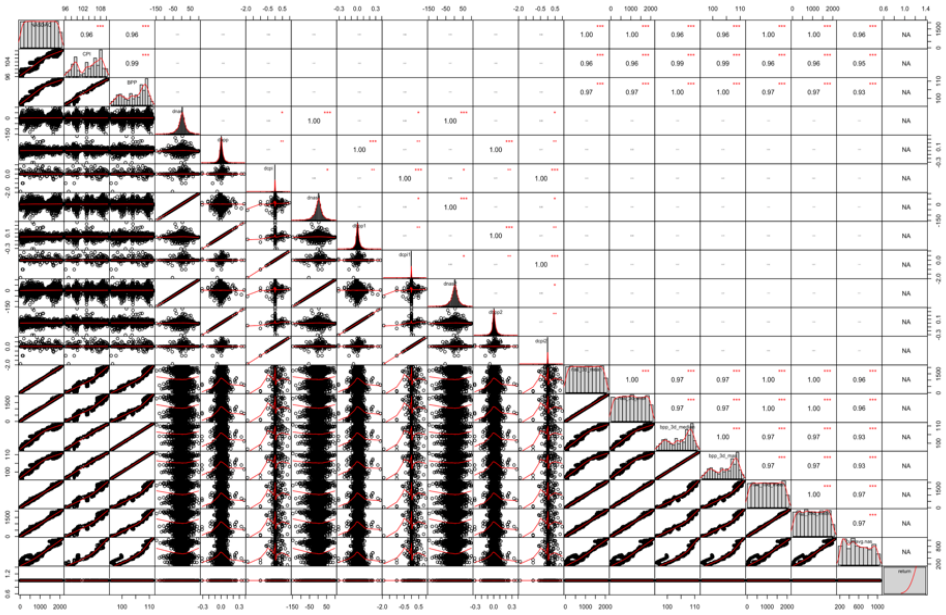
The goal of this project was to first use time-series NASDAQ data and inflation indicators in a model to predict future performance of the NASDAQ index, and then select an investment strategy that would guide us to buy or not buy our holdings in the NASDAQ index based on the predicted values from our model.

Data Processing and Feature Engineering

Our first step was to visualize the data and convert it into a time series (data exploration can be found in Appendix A). Next, we performed feature engineering. Our models mostly use the changes in NASDAQ, BPP and CPI to predicate the NASDAQ (or change in NASDAQ), and also consider the effect of the short-term, medium-term and long-term shocks.

- Short-term shock variables: The rolling mean, median, and maximum of the financial measurements with a window of three observations (not including current day)
- Medium-term shock variables: The rolling means, medians, and maximums over a 10-day and 20-day window (not including current day)
- Long-term shock control variable: cumulative means (not including current day)

The full list of created features is as follows: first differences of NASDAQ, BPP, and CPI; lag-1, lag-2, and lag-3 features for all three first differences; NASDAQ 3-day rolling mean and max; BPP 3-day rolling mean, median, and max; NASDAQ 10-day rolling mean, NASDAQ 20-day rolling mean; NASDAQ cumulative mean. We explored the correlations between all the variables that we created:



There is a strong correlation between variables (NASDAQ, BPP, and CPI) that are not differenced (visible in the top left corner of the graph), but after taking first differences, these variables are essentially correlated only with their own lags.

To obtain our final dataset, we dropped all rows with NA values. In order to choose the model that would give us the best out-of-sample performance, we trained our models on a training set composed of the first 1500 observations in the data (75% of the sample data). To choose the best investment strategy for each model, we used a validation dataset composed of the final 626 observations. We evaluated our model's and investment strategy's performance on this validation set as a proxy for out-of-sample performance.

Model and Investment Strategy Selection Process

The models we tried can be divided into five main categories: linear regression model, support vector machine, Lasso, Random Tree, and Extreme Gradient Boosting model, and all of the five models are used to investigate the changes in the NASDAQ Composite Index (i.e., the difference between the previous period and this period for the NASDAQ = 'dnas'), which is almost white noise. See Appendix B for additional discussion on model type selection.
For the perspective of the length of the lags included in the models, our models could also be divided into short-term models, medium-term models, and long-term models. The short-term models contain variables with one-period lags: dnas ~ dnas1, and dnas ~ dnas1 + dbpp1 + dcpi1 controlling for the short-term effect of BPP and CPI. The medium-term models contain variables with two- or three-period lags: dnas ~ dnas1 + dbpp1 + dcpi1 + dbpp2 + dcpi2, dnas ~ dnas1 + dbpp2 + dcpi2, and dnas1+bpp_3d_median. The long-term models contain all the lagged variables: dnas ~ . -dnas-NASDAQ-return-dbpp-dcpi-CPI-BPP.

Alongside these models, we also considered three investment strategies:

1. The first investment strategy is called Best-quantile Investment Strategy or the quantile method: the strategy would help to find an investment range with the upper bound and lower bound (e.g., $[a, b]$), and then buy the NASDAQ Index when the predicted 'dnas' is in this region **(**$a \leq {}'dnas' \leq b$**).** The reasoning behind this approach is that if a model predicts a very high value of dnas, it may be an unreliable prediction, so we want to take a conservative approach by excluding very high percentiles of predicted dnas. Specifically, the procedure of this investment strategy to find the upper and lower bounds in the following way:

1.1. Get a series of predicted values of dnas from our models.

1.2. For each percentile p from 0-50, let the p^{th} percentile of the predicted values of 'dnas' be the lower bound for investment. For each percentile q from 51-100, let the q^{th} percentile be the upper bound for investment. For every pair (p,q), calculate the IR, average daily return, and total return if our investment strategy is buying NASDAQ when the predicted dnas is between the p^{th} and q^{th} percentiles.

1.3. Choose the pair (p,q) that gives us the highest IR as the final percentile values. Select the value of dnas that corresponds to the p^{th} percentile as the lower bound and select the value of dnas that corresponds to the q^{th} percentile as the upper bound for the optimal investment strategy.

2. The second investment strategy is called Best-threshold Investment Strategy or the threshold method. This strategy seemed to be the best and most robust. Our Best-threshold strategy would find a specific threshold T within the range [-20, 10] (here we set the step is 0.1, i.e., seq(-20, 10, by = 0.1)). When the predicted 'dnas' is greater than the threshold T, the Best-threshold Investment Strategy would buy the NASDAQ Index, otherwise no purchase would be made. Our algorithm would test each point in the list [-20, -19.9, -19.8 ,…., 9.9, 10] as a potential threshold, and return a list of IRs, average daily returns, and total returns under the candidate thresholds. The threshold corresponding to the highest IR is chosen as our optimal threshold for the out-of-sample test.

3. The third investment strategy is called the Rolling-average Investment Strategy:
The reference for this investment strategy is the average return over a period of time. In order to comprehensively take into account the short, medium and long term effects, this method calculates the NASDAQ average of the previous 5, 10, 20, 30, 50, and 100 periods, notated as $m_i$for window length i. Then if the predicted 'dnas' value falls below the m_i minus 1 standard deviation, we buy the NASDAQ index, and if the predicted 'dnas' value falls above the m_i plus 1 standard deviation, we sell. The reason is that we want to buy low and sell high. Last, we compare all the IRs, average daily returns, and total returns corresponding to each m_i, and choose the window length i (and its corresponding m_i) with the highest IR as our final values. This optimal $m_i$and its standard error are our reference indicators for out of sample investment.

See Appendix C for additional discussion of investment strategy selection.

We chose to evaluate our models both on accuracy and on investment performance. Since the models that we created are pretty different, some models like Random Tree and Xgboost would perform better in accuracy, but might have an overfitting problem. Other models like linear regression could be effective in prediction but have relatively lower accuracy, so we finally chose to use investment metrics like IR, average daily return, total return and accumulated cash value to evaluate our models. We compared our models' investment metrics to metrics from the baseline strategy of buying NASDAQ in every period.

Below are detailed descriptions of our top candidate models and investment strategies.

Top Models and Investment Strategies

We first considered five linear regression models with different inputs. Each model tried to predict 'dnas' (change in NASDAQ) from the variables listed below:

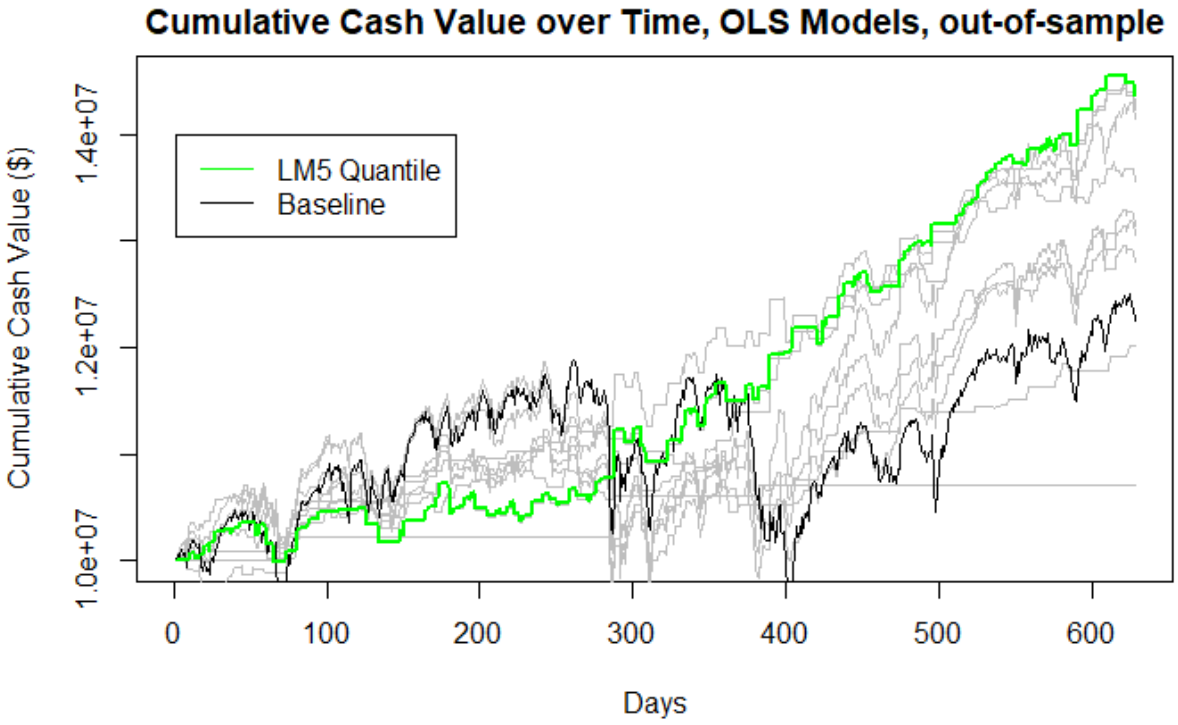
1. LM1: dnas1
2. LM2: dnas1 + dbpp1 + dcpi1
3. LM3: dnas1 + dbpp1 + dcpi1 + dbpp2 + dcpi2
4. LM4: All prior known features
5. LM5: dnas + bpp_3d_median

Using domain knowledge we suspected that these models would perform the best. However, we also implemented a lasso model which tends to do feature selection automatically. The results for each of these models are given below for both the quantile and threshold investment strategies. NOTE: results may vary from attached code due to randomness.

	Quantile Method			Threshold Method		
Model	IR	Mean Return	Cash Return	IR	Mean Return	Cash Return
LM1	0.126	0.000573	0.432	0.0544	0.000460	0.305
LM2	0.117	0.000579	0.427	0.0553	0.000422	0.276
LM3	0.112	0.000494	0.427	0.0553	0.000422	0.280
LM4	0.102	0.000296	0.201	0.0612	0.000112	0.071
LM5	0.128	0.000587	0.436	0.0613	0.000463	0.313
Lasso	0.037	0.000375	0.226	0.0374	0.000375	0.226

There are several limitations to the linear regression methods. First, if the relationships between these variables are not linear, these models would give poor results. Second, there are many possible combinations of features so determining which ones are the best features to use in modeling is a difficult task. Lastly, the Lasso model sets all of the coefficients to 0 and always predicts the intercept - the average return over the training data. Predicting dnas is a challenging problem, and it is hard to obtain strong predictors.

Below is a chart of cumulative cash value over time for each of the OLS models (the Lasso model is identical to the baseline). Here, out-of-sample refers to our validation set.



The best model appears to be LM5 with the quantile method as the investment strategy. The quantile method applied to LM5 tells us that we should invest when the predicted return from LM5 is between 2.66 and 3.91 to get the best return.

After getting the results from the linear models, we decided to experiment with some non-linear models. We started with five SVM models using a similar variable selection process as we did with the linear models. The details of these five models for predicting dnas (change in NASDAQ) are given below:

1. SVM1: dbpp2 + dcpi1 + nas_3d_mean

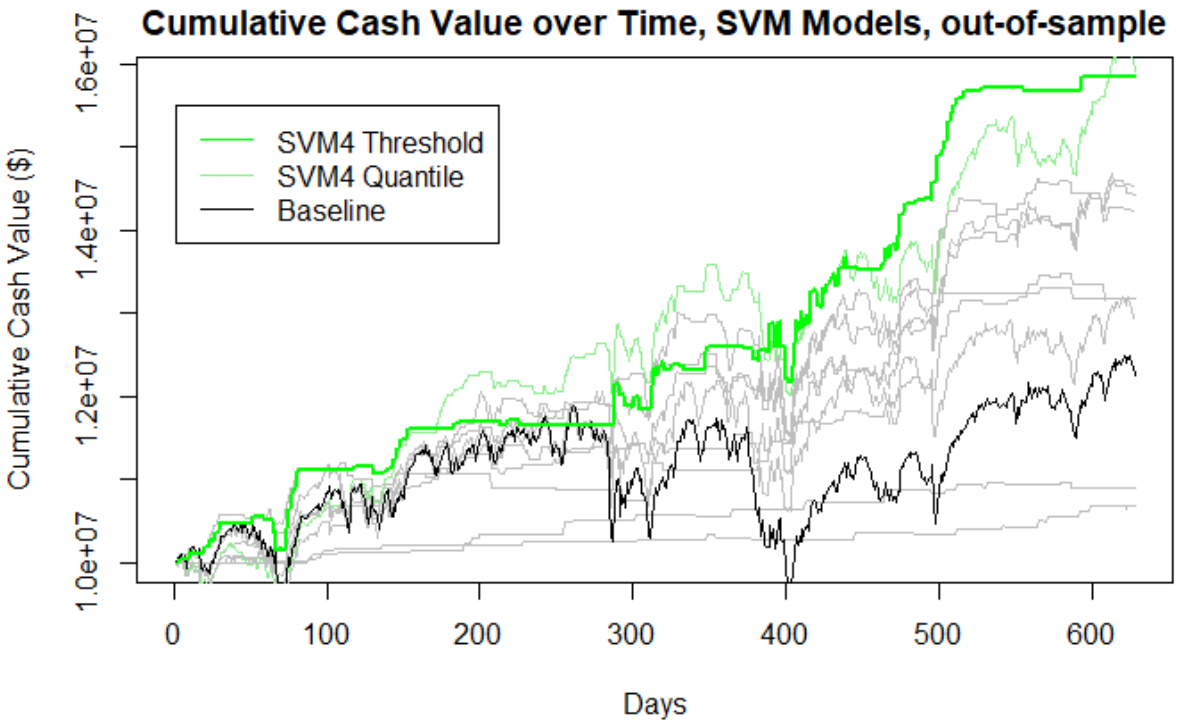
2. SVM2: dnas1 + dbpp1 + dcpi1+ dbpp2 + dcpi2
3. SVM3: All prior known features
4. SVM4: dnas1 + dcpi1 + dnas2 + nas_3d_mean + nas_3d_max + bpp_3d_median + nas_10d_mean + cumavg.nas
5. SVM5: dnas1+ dbpp1+ dcpi1+ dbpp2 + dcpi2 + nas_3d_mean + nas_3d_max + bpp_3d_median + bpp_3d_max + nas_10d_mean + cumavg.nas

In addition to these models we wanted to try some more complex non-linear models, so we implemented random forest and XGBoost models. The results for these models are given below. NOTE: results may vary from attached code due to randomness.

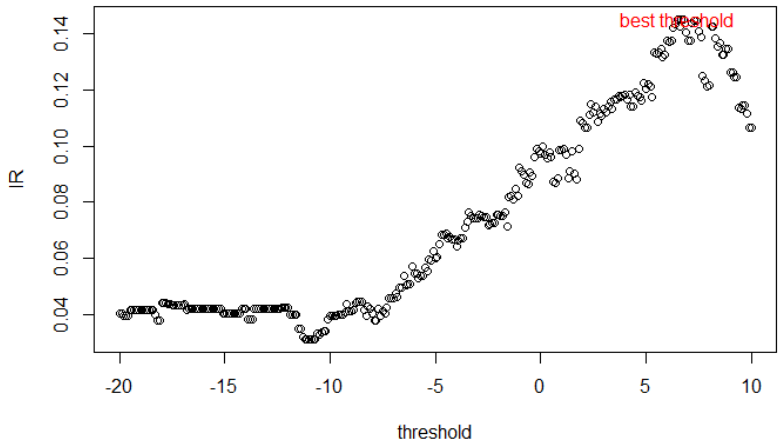
	Quantile Method			Threshold Method		
Model	IR	Mean Return	Cash Return	Quantile Method	Threshold Method	Rolling Method
SVM1	0.109	0.000449	0.318	0.0873	0.000586	0.423
SVM2	0.112	0.000137	0.089	0.0489	0.000459	0.295
SVM3	0.081	0.000613	0.443	0.0887	0.000623	0.454
SVM4	0.010	0.000771	0.591	0.145	0.000748	0.585
SVM5	0.118	0.000104	0.067	0.103	0.000451	0.318
Random Forest	0.089	0.000557	0.400	0.0538	0.000381	0.250
XG Boost	0.115	0.000498	0.358	0.041	0.000396	0.242

These models do not have the same linearity assumptions as the previous set of models, however they may still have some limitations. The models are more complicated so they are more likely to overfit on the training data. Additionally, the models are less interpretable. Given the context of the problem interpretability is not a necessity, but it would still be good to explain why the market behaves in certain ways.

Below is a plot of cumulative cash value over time for the SVM models. Similar plots for our random forest and XGBoost models can be found in Appendix D. Again, out-of-sample here refers to our validation dataset.



From the plot and table above, we see that the best method from this category of models is SVM4 with the threshold method. The threshold method applied to SVM4 tells us that we should buy when the predicted value of dnas from the SVM4 model is above 6.5, as we can see from the plot below (the IR is highest at the threshold of 6.5).



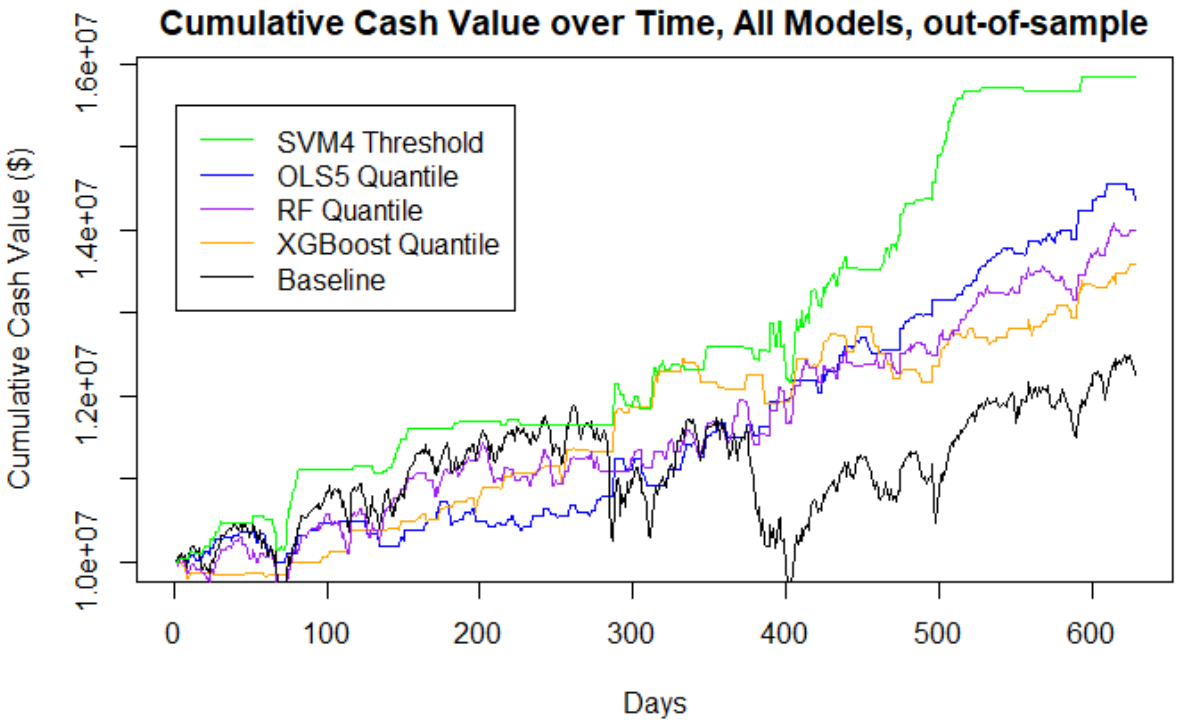
Final Model Selection

We performed our final model selection by comparing our best SVM model, our best OLS model, our random forest model, and our XGBoost model. As before, we fit the models to our

training set of 1500 observations and selected an investment strategy using the validation set of the last 1500 observations. Below is a table comparing information ratio, mean return, and cash return on the validation set for all these models.

	InformationRatio	MeanReturn	CashReturn
SVM4_threshold	0.14524876	0.0007484115	0.5853872
OLS5_quantile	0.12819475	0.0005872815	0.4356151
xgboost_quantile	0.11468755	0.0004975903	0.3580678
RandomForest_quantile	0.08944385	0.0005565333	0.4003977
baseline	0.03742867	0.0003750008	0.2258156
lasso_quantile	0.03742867	0.0003750008	0.2258156

We also plotted cash holdings over time for each model:

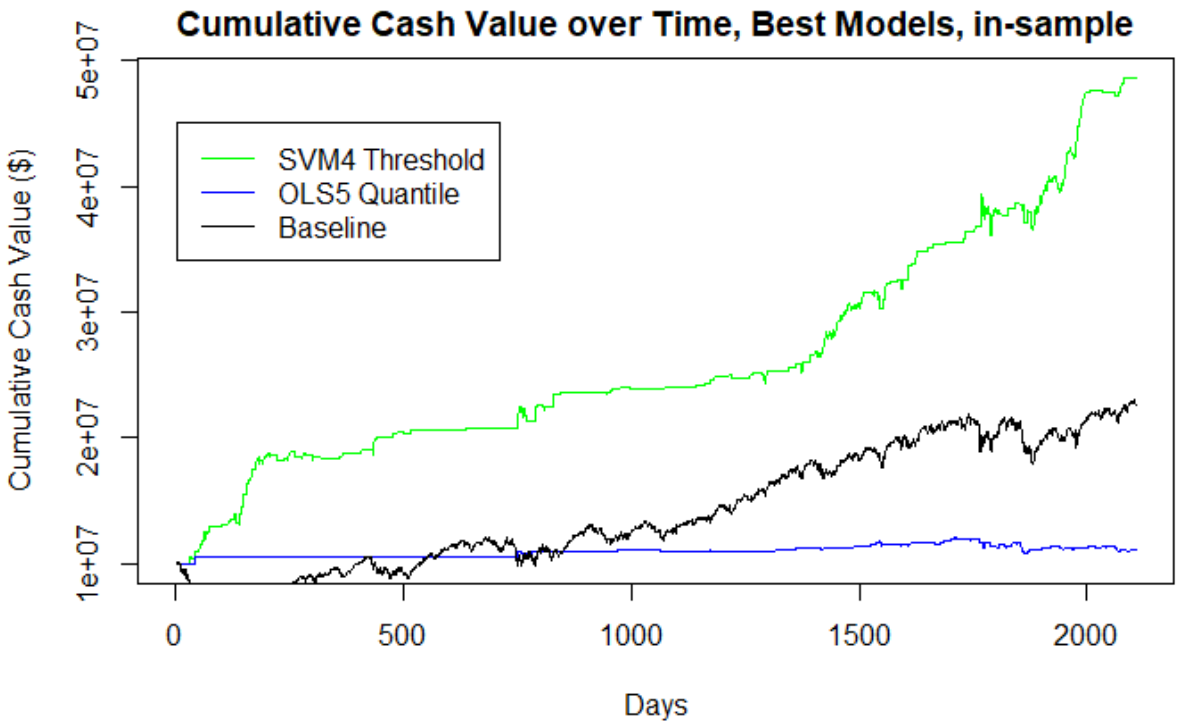


Finally, we compared model performance by calculating R^2 on the validation set:

Model <chr>	OutOfSampleRSquared <chr>
Best SVM	0.017364390833
Lasso=Baseline	-0.000009669066
XGBoost	-0.000461135132
Best OLS	-0.005734720123
Random Forest	-0.237597417704

The best SVM model (predicting dnas from dnas1 + dcpi1 + dnas2 + nas_3d_mean + nas_3d_max + bpp_3d_median + nas_10d_mean + cumavg.nas, with dnas investment threshold of 6.5) and the best OLS model (predicting dnas from dnas1 and bpp_3d_median, with an investment threshold of predicted dnas between 2.66 and 3.91) have higher IR, mean return, and cash return than all other models and the baseline. In addition, the SVM model has better predictive power than all other models. We select these two as our semi-finalist models.

Next, we re-trained the SVM model and the OLS model on the entire dataset, made predictions on the entire dataset, and then calculated metrics based on their predetermined investment strategies. Below is a plot of the cash value over time for the two models and the baseline:



Below are final IR, Mean Return, and Cash Return values for the models and the baseline, trained on the entire dataset and making predictions on the entire dataset:

Model <chr>	InformationRatio <dbl>	MeanReturn <dbl>	CashReturn <dbl>
SVM4Threshold	0.11956171	7.720039e-04	3.8645314
OLS5Quantile	0.01760183	5.365724e-05	0.1088035
Baseline	0.03425045	4.869876e-04	1.2532188

We can see that our SVM model clearly outperforms the OLS model and the baseline, indicating that the OLS model's investment strategy probably overfitted to the validation data. We choose

the SVM model with its specific investment strategy as our best model. Below is a final summary:

Best model	svm(dnas ~ dnas1 + dcpi1 + dnas2 + nas_3d_mean + nas_3d_max + bpp_3d_median + nas_10d_mean + cumavg.nas)
Investment strategy	Invest if predicted dnas is over 6.5
In-sample IR	0.11964
In-sample Mean Return	0.000773

Appendix A: Data Exploration

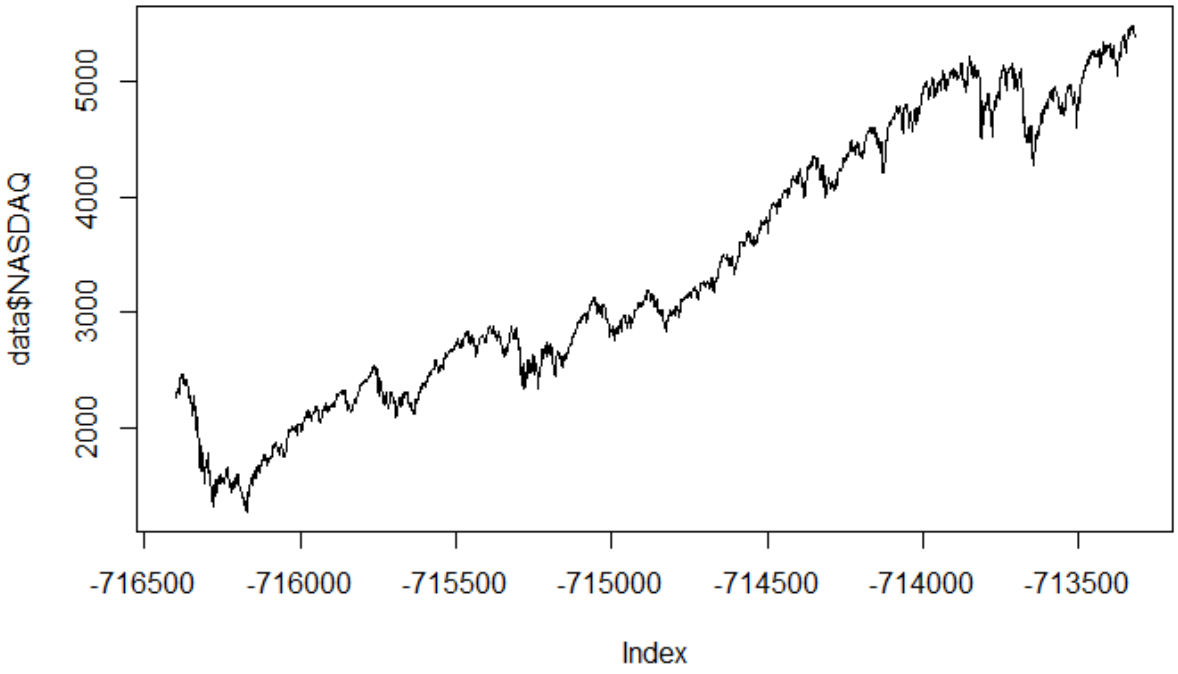


Figure 1: NASDAQ over time

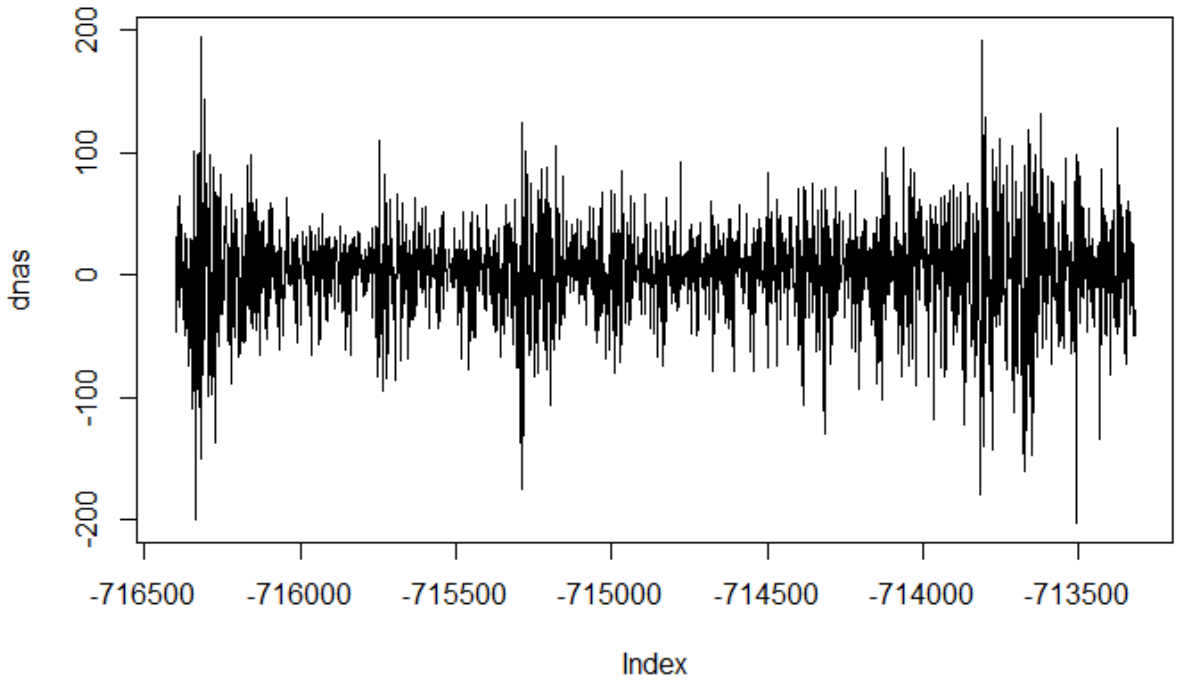


Figure 2: NASDAQ first differences (dnas) over time

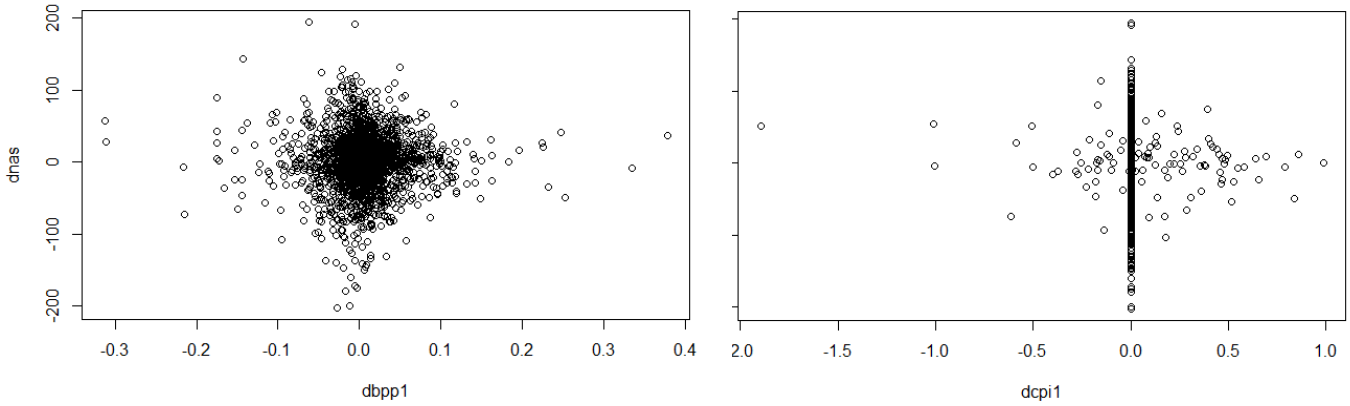


Figure 3a (left): NASDAQ first differences vs. lag-1 BPP first differences

Figure 3b (right): NASDAQ first differences vs. lag-1 CPI first differences

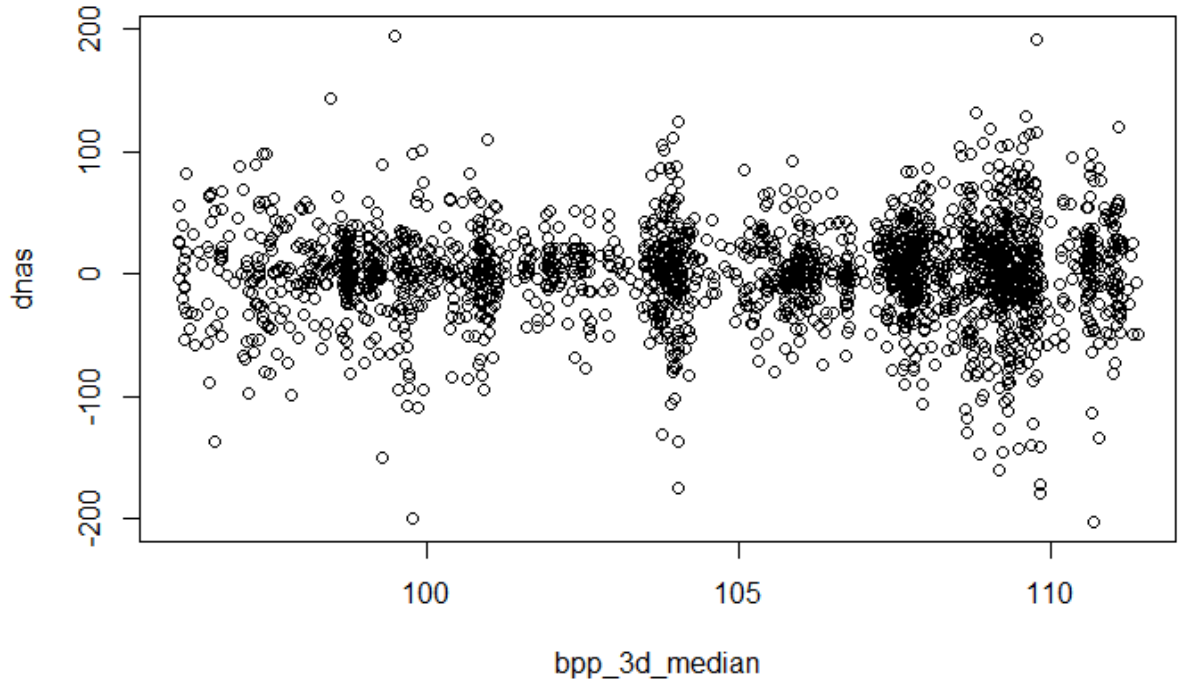


Figure 4: NASDAQ first differences vs. BPP rolling 3-day median

Appendix B: Discussion of Modeling Choices

(1) The linear regression model mainly explores the relationship between the change in NASDAQ Composite Index at this period and its lagged one period (dnas1), after controlling some external shocks. Since the previous change in the NASDAQ, i.e., dnas1, explains the current change in the NASDAQ, i.e., dnas to a large extent, so except dnas1, most other variables in the linear regressions are statistically not significant, thus we need other models to investigate other important variables besides dnas1.

(2) Lasso model: Lasso is a good way to find important variable without overfitting problem, in our Lasso model, we consider these variables: 'NASDAQ', 'CPI', 'BPP', 'return', 'dbpp', and 'dcpi'. But we found out that the performance of the Lasso model even did not beat the simple linear regression without any control (i.e., dnas~dnas1). The major reason is that lasso picked none of the variables as significant and pushed them all to zero. The predicted value is the intercept. Although Lasso might give us meaningful results if we set the right random seed, we have to try hard to select a specific seed which implies that this method is not very robust and possibly would not generalize well.

(3) The support vector machine (SVM): part of the reason for setting this model comes from the investment approach - our investment approach is just to decide whether or not to buy the NASDAQ index (based on the predicted dnas), in other words, we can classify the dnas data into two categories. One is the buy category and the other is the don't buy category. Since SVM model performs well on categorical data, we consider using the SVM model.

(4) Random Tree: although, in linear regression model and Lasso, the predictive effects of 'NASDAQ', 'BPP', 'CPI', 'return', 'dbpp', and 'dcpi' are significant enough, in some specific domain of these variables, it is still possible that these variables effectively predict ‘dnas’, so that we try to use random tree in finding a better model.

(5) Extreme Gradient Boosting model (Xgboost model): Xgboost model is close to the random tree. Both are decision trees, but random forest builds each tree independently, while Xgboost builds one tree at a time. The advantages of the Xgboost model are that the prediction error given by Xgboost is lower than that of Random Tree, and Xgboost strengthens the model with weak predictions. Those features are exactly what we need, so we also build the Xgboost model.

Appendix C: Discussion of Investment Strategy Methodology

The methodology for selecting the optimal investment strategy for each model could be divided into four main steps:

Step 1: Fit the model with the in-sample data and get the key elements of each model such as parameters, classification ways, etc.

Step 2.1: Forecast the model with the test data and get the predicted values of dnas.
Step 2.2: Use the predicted values and the corresponding actual values to get references of each investment strategies: for example, to get Best-quantile Investment Strategy's optimal upper and lower bounds, the Best-threshold Investment Strategy 's threshold, and Rolling-average Investment Strategy's average daily return m_i and its standard error.

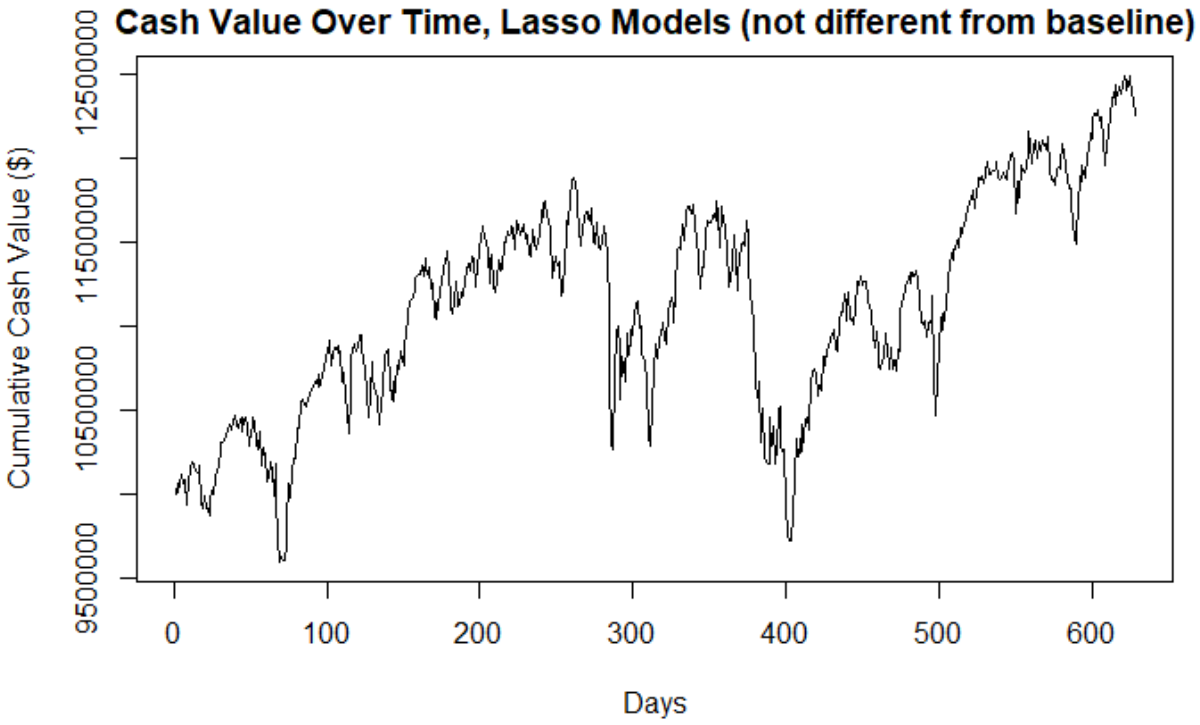
Step 3: Use the validation sample to simulate a real investment. Take linear regression as example, since we already have the optimal investment range from the second step and we can get the predicted value based on the model, then only if the predicted values are in the investment range, we buy the NASDQ index, and we can also calculate the IR, average daily return, and total return based on the Best-quantile Investment Strategy. Doing the similar work for all other investment strategies and models. In the end, we get all the IR, average daily return, and total return for all different models under the three investment strategies.

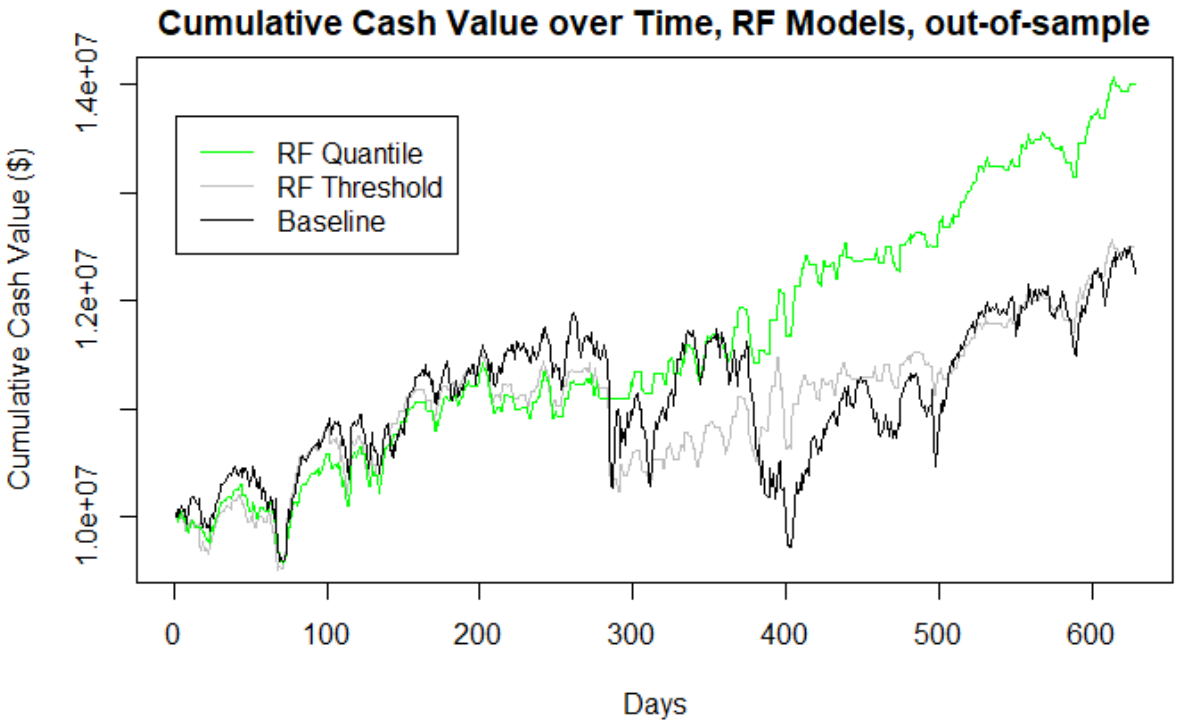
Step 4: We compare the returns of the same model under different investment strategies and select the best performing investment strategy for each model.

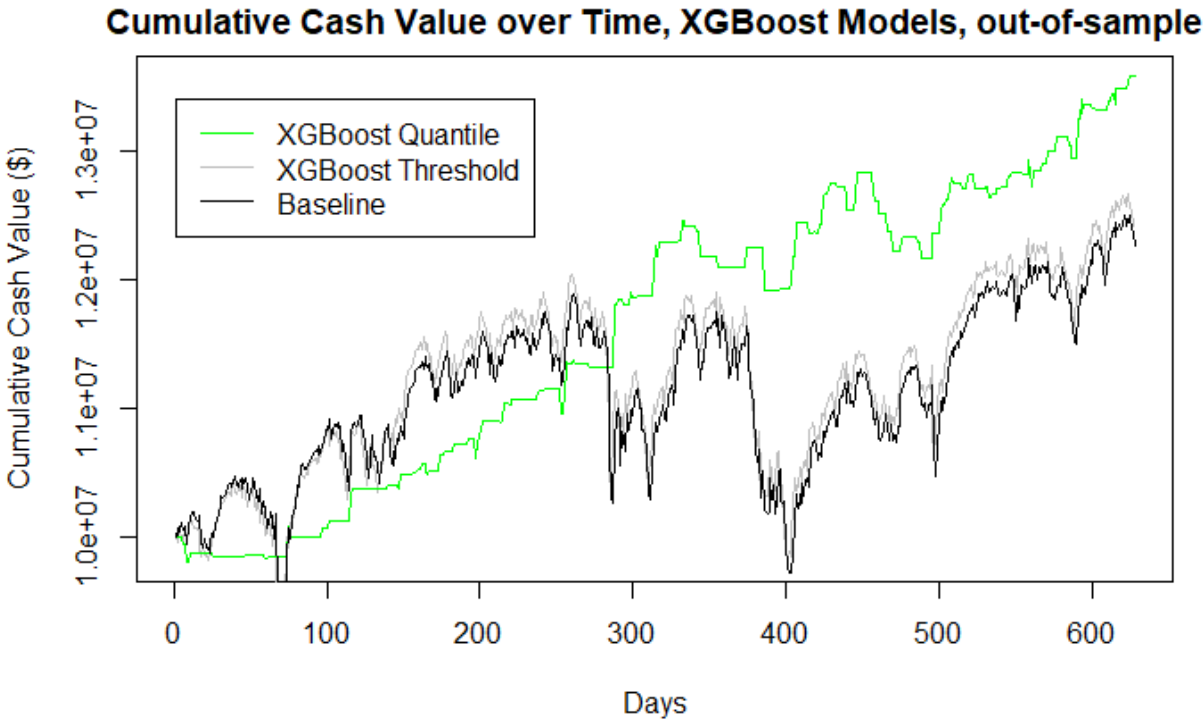
One Caveat: since the test sample (only 126 observations) is not large enough for other models except linear regression models, so we use the out-of-sample data (combining test sample and validation sample) for steps 2 and 3. Although this would return higher IRs, average daily returns, and total returns than the stringent out-of-sample test, fortunately, the linear regression born under the stringent conditions has almost the best performance in this game, especially taking the Best-quantile Investment Strategy. Although linear regression is slightly behind SVM under the Best-threshold Investment Strategy, probably because SVM uses a soft margin which loosens the threshold.

One reason for not reducing the amount of in-sample data is that although the ‘dnas’ of the first 500 observations are also close to white noise, the variance of the returns of these 500 observations are much larger than the variance of the returns of the next 1600 observations, in order to promise the robust of our model, we do not reduce the amount of in-sample data.

Appendix D: Additional Plots, Cash Value Over Time




Cumulative Cash Value over Time, RF Models, out-of-sample
RF Quantile
RF Threshold
Baseline
Cumulative Cash Value ($)
1.0e+07
1.2e+07
1.4e+07
0
100
200
300
400
500
600
Days


Cumulative Cash Value over Time, XGBoost Models, out-of-sample
XGBoost Quantile
XGBoost Threshold
Baseline
Cumulative Cash Value ($)
1.0e+07
1.1e+07
1.2e+07
1.3e+07
0
100
200
300
400
500
600
Days