CS 7464 MC3P3 Report

lyang338

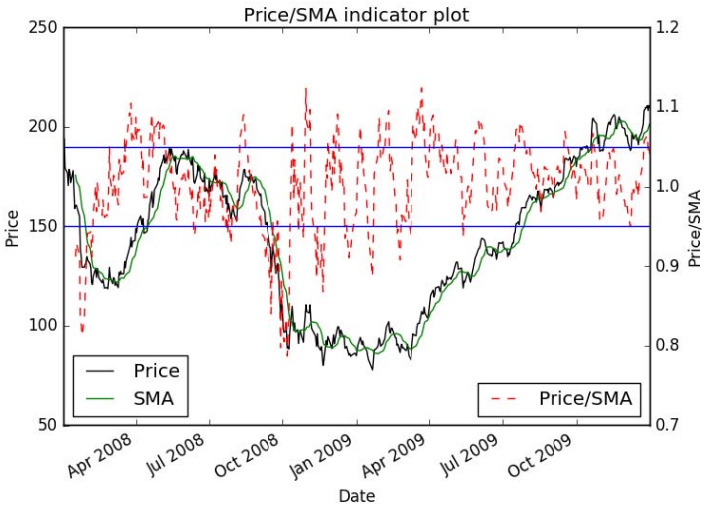
Overview:

In this project, we will combine what we have learned so far about market simulator, random tree learner, bag learner as well as knowledge of pandas implementation, to model, train and test a learning trading algorithm.

Part 1: Technical Indicators

The first thing we need to do for this project is to do some research to find some appropriate technical indicators for our trading algorithm. There are a lot of resources on line about these indicators. Here I will discuss the three indicators I will use.

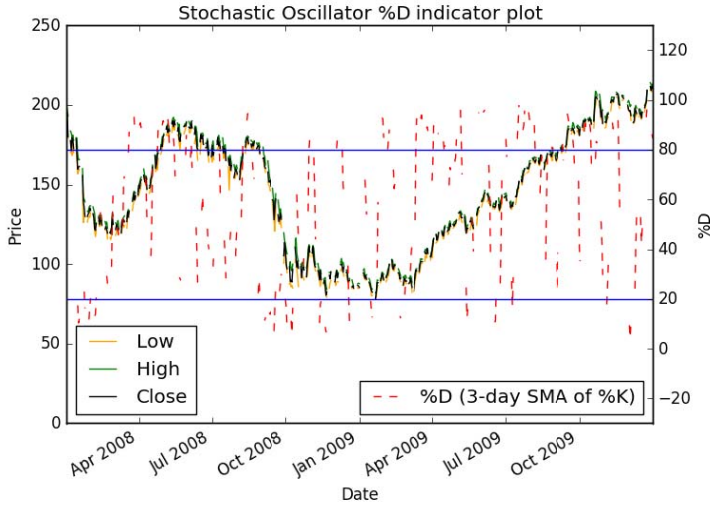
The first one I will choose has been discussed by professor in the lecture, which is Price/SMA ratio. A simple moving average (SMA) is just an arithmetic moving average during a specific number of time periods (I will 12 for my implementation), and it is the most common type of average used by technical analysts. According to the lectures, when the Price/SMA ratio is higher than 1.05, then it is a signal of overbought, and when it is below 0.95, usually it is a signal of oversold. For the implementation of this project, since we don't want some none values for the several beginning days, what I do is to choose a longer time range for calculation first, then select/slice the data frame for required time range. The same idea will be used for all other indicators. Below is the plot of Price/SMA indicator. The left y-axis is used for Price and SMA lines, and the right y-axis is used to the ratio plot. I didn't normalize the ratio here (which will be done for the later machine learning part), and from the two horizontal lines (y=0.95 and y=1.05), we will have an abstract idea of overbought and oversold signals related to the Price/SMA ratio.



The second indicator I will use is Stochastic Oscillator %D indicator. To introduce the %D indicator, we need to discuss the %K indicator first. The Stochastic Oscillator %K measures the level of the close relative to the high-low difference during a given period of time (I use 14 for the time period according to some blog). The equation is for %K is as below:

$$\%K = (Current\ Close - Lowest\ Low)/(Highest\ High - Lowest\ Low) * 100$$

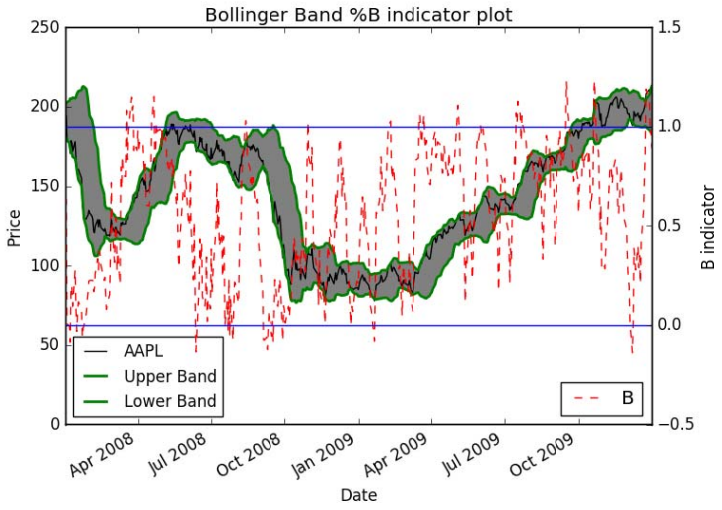
And the %D indicator is the 3-day SMA of %K. For the interpretation of it, usually a relatively low reading (like 20) would indicate a oversold signal, and high reading (above 80) most time means the price is too high. Below is the plot for my implementation result. The left y-axis is for the prices (Low, High and Close), and the right y-axis is the %D indicator. From this plot we can also get an abstract idea of trading signals.



The last indicator I will use has also been discussed by the professor. It is the Bollinger Band %B indicator. We have already learned how to calculate the upper and lower band for Bollinger Band in the lecture, and the %B indicator equation is just as below:

$$\%B = (Price - Lower\ Band)/(Upper\ Band - Lower\ Band)$$

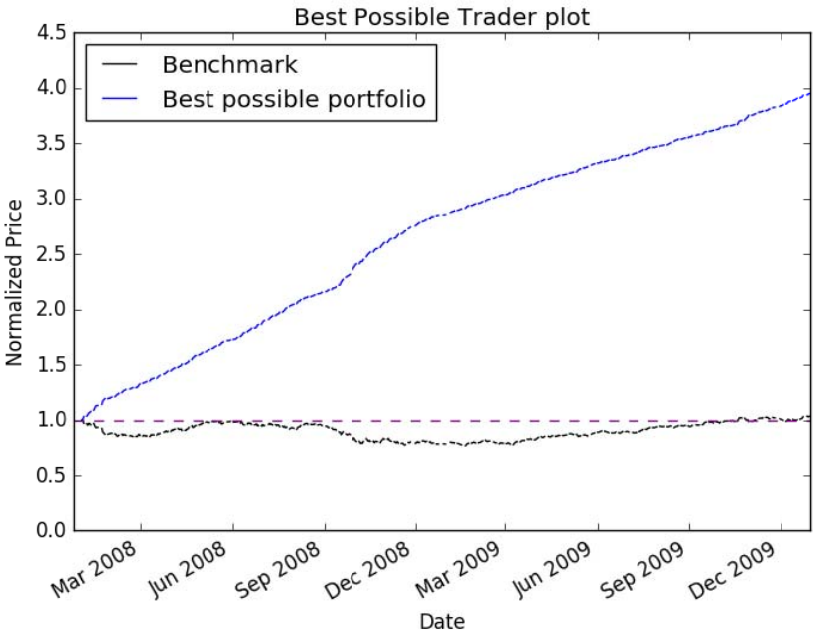
As discussed in the lecture, when the %B indicator is higher than 1, then it means that the price is too high, and when the %B is lower than 0, it usually indicates an oversold signal. Below is my plot for the Bollinger Band %B indicator. The two green lines are the upper and lower bonds of the Bollinger Band, and I also use the grey color to fill to the band. The right y-axis is used for the %B indicator. By both the grey Bollinger Band and %B indicator, we can have a straight idea of overbought and oversold signals.



Part 2: Best Possible Strategy

In the part, we need to design a strategy to see the best possible trader. In order to match and compare to the following discussion is part 3 and 4, here we limit our position to 200 (long or short). Assume we could see the future, the best strategy would be very simple: when the stock price will increase tomorrow, then we should 'LONG', and if it decreases, we should 'SHORT'. Every day we should close the position first, then decide 'BUY' or 'SELL', in order to keep our 200 limit position. In the implementation, first I will use the best trading strategy to create an best order list, then write the list to a csv file. Then I will use my 'masketsim.py' (same as in code MC2P1) process the saved order file. For the benchmark portfolio, I will manually create a csv file, which just contain one 'BUY' at the beginning (2008-01-02) and one 'SELL' at the end (2009-12-31), and then process the order in the same way. Below is the plot of the comparison of the benchmark portfolio and best possible trading strategy portfolio. Also I have listed the information of cumulative return, stdev of daily returns as well as mean of daily returns in a table. As we can see, the cumulative return is really high, and from the plot and table we will have an idea of the upper limit of trading performance.

	benchmark	best strategy
cumulative return	0.03164	2.9495
stdev of daily returns	6.776E-5	0.002699
mean of daily returns	0.008283	0.003281



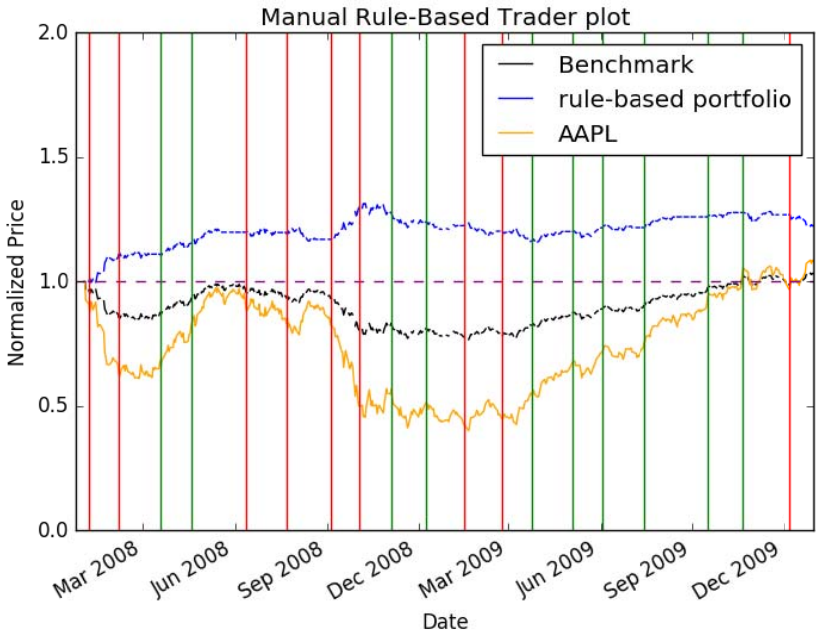
Part 3: Manual Rule-Based Trader

Now with the best possible cumulative return in mind, we should move forward to design some real trading strategy. In this part, we will use these technical indicators we have built in part 1, to create an manual rule-based trader. In my understanding, this is more like a traditional trading strategy, which we need to understand and extract some indicators, then create a trading model based on the values of the indicators. In the part 1, we have already discussed the three technical indicators and their signals to trading. Here I will list them as a table so we can have a better understanding.

Indicator	'overbought' signal	'oversold' signal
Price/SMA ratio	> 1.05	< 0.95
Stochastic Oscillator %D	> 80	< 20
Bollinger Band %B	> 1	< 0

With those methods in part 1, we can easily calculate these indicator values. For all these values, one particular important thing is the choice of period window. It is clear that different window size will result different indicator values, which will further lead to different trading strategy. In the real world, traders will carefully choose these window period and signal threshold values for each technical indicator based on their knowledge and experience. In this project, I have tried several popular period values and finally I use 12 for Price/SMA, 14 for %D and 20 for %B. For the threshold values, I just pick those popular ones as in the above table.

As discussed in the lecture videos and notes, the simplest strategy is that when all the indicators signal 'overbought', we will 'SELL', and 'BUY' when all indicate 'oversold'. This is a quite easy and safe way. However, since here we have been bound by the 21 trading day hold, if we need to wait all three signals come together, there will be very few trading opportunity for us. Therefore, in my rule-based strategy, I have used '2/3' rule, which means when two indicators indicate the same signal, then I will process the trading. Also in the implementation, whenever we trigger a 'LONG' or 'SHORT' trading, we should wait for another 21 days (except that close the end period, we need to check this during our implementation). Besides, we also need to close the position at the final day (12-31-2009) for both benchmark and our rule-based trader. Below is the plot for the comparison of benchmark and our manual rule-based trader performance. Note these vertical green and red lines indicate 'LONG' and 'SHORT' entry points respectively. Also I have listed the performance information in the table.



	benchmark	rule-based trader
cumulative return	0.03164	0.2251
stdev of daily returns	6.776E-5	1.752E-4
mean of daily returns	0.008283	0.005636

From the above plot and the table, we can see that the rule-based trader works well. The cumulative return is about 22%, while the benchmark is only about 3%. Also we can see that around 2008-09, there are several 'SHORT' trades, and considering the price drop around that period, these trades are really good. The same thing happens around 2009-06, when the price continues to increase, my manual rule-based trader also performs well then. Overall the rule-based trading strategy works well and I believe with more domain knowledge of technical indicators and trading experience, the performance will be much better.

Part 4: ML Trader

After finishing the traditional rule-based manual trading strategy, now we will move forward to the new machine learning trader. Recall that in the previous project, we have already successfully built a random decision tree learner and a bagger learner, also we have seen the effect of various leaf size of the tree and various number of the bags. In this work, we will construct a ML trader based on these previous work.

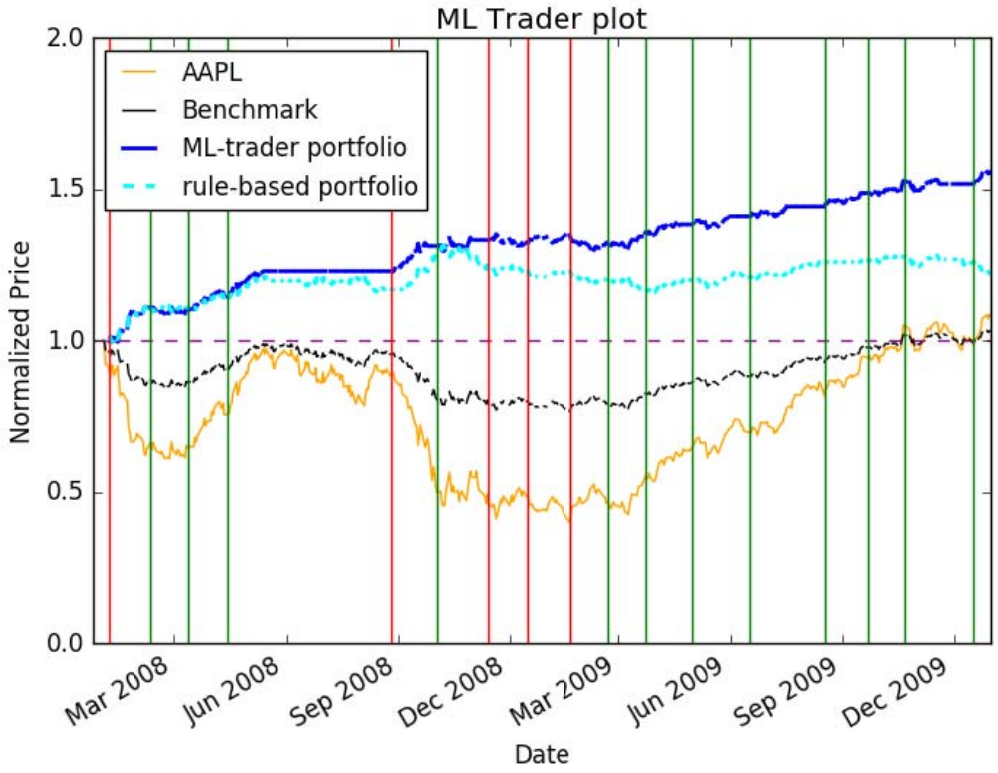
The first thing we need to do a machine learning work is always the feature selection and label definition. For the features, we will use the three technical indicators. However, unlike the previous, here we will need to normalize the three indicators so that they will have zero mean and standard deviation of 1.0, which will be helpful for our machine learning train and test. For the label, we will define two threshold values 'Ybuy' and 'Ysell'. We can calculate the 21-day returns based on our data, and if the return is bigger than the 'Ybuy', then we will buy it, or label it as 'LONG' or '+1'. Similarly, if the return is smaller than 'Ysell'. we will label it as 'SHORT' or '-1'. If the return is between the 'Ybuy' and 'Ysell', then the label should be 'DO NOTHING' or '0'. In the implementation, it is always good to test several pairs of 'Ybuy' and 'Ysell' to see the final performance. After some trials, finally I choose 'Ybuy' as 1.1 and 'Ysell' as 0.9. Once we have those threshold values, we can label all the days in the period and save them along with their normalized feature values to a csv file. Recall that in the previous homework, we are provided with these labeled csv files.

To avoid overfitting problem, we will set our leaf size to be bigger than 5 (I use 6). Also I will use bagger learner to increase my performance. For the bagger learner, it is necessary to choose a good bag number. By varying the bag number from 0 to 100, and then calculate the RMSE, it shows that the good bag number is above 50 (and I choose 60) for my implementation. For the learner, we will use the whole data period for both train and test, and the output from the prediction is just a list of labels (totally 505). Since we have used bagger learner, the outcome prediction labels are no longer (-1, 0, 1) anymore due to average. Thus, we may need to deal with this during our order processing.

Now with our test prediction result, we need to use it to create an order, which will be executed later. As we have mentioned above, the output label is no longer integers but float numbers between -1 and 1. Hence we need to have a way to transfer them to a 'LONG' or 'SHORT' decision. In my implementation, I will use threshold values -0.5 and 0.5 to make these decision (this will be varied based on different situations). The simple decision table is as below.

Prediction value	Decision
> = 0.5	LONG
<= -0.5	SHORT
other	DO NOTHING

With the above decision table, we can easily generate an machine learning based trading order, and then execute the order and calculate the performance. The result is in the plot and table below.



	benchmark	rule-based trader	ML trader
cumulative return	0.03164	0.2251	0.5554
stdev of daily returns	6.776E-5	1.752E-4	8.089E-4
mean of daily returns	0.008283	0.005636	0.004894

As we can see from the above plot and table, the performance of machine learning (typically random decision tree learner and bagger learner) is much better than manual rule-based trading strategy. Especially during the period around 2009-06 - 2009-12, when the price of stock goes up, the machine learning trader has successfully made several good trades. The work of this part is quite interesting, since it gives us a straight idea how machine learning will used to improve the performance. Also, although it seems that the machine learning trader doesn't care about traditional models, the domain knowledge of finance is still important to get a good working ML trader, since we need to use these knowledge to carefully pickup the useful features (and I believe under different situations, the feature selection would be quite different).

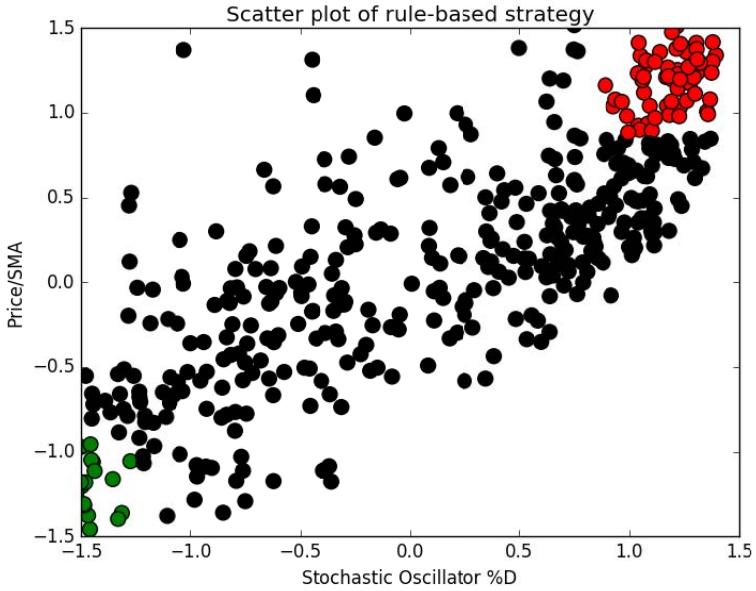
Part 5: Visualization of date

For this part, we will perform the scatter plot for rule-based strategy and machine learning strategy, in order to gain a better view of these trading strategies.

First we will choose two indicators, the two I will use are Price/SMA ratio and Stochastic Oscillator %D value. Again we can refer for the table below for details about the trigger signals of these two technical indicators.

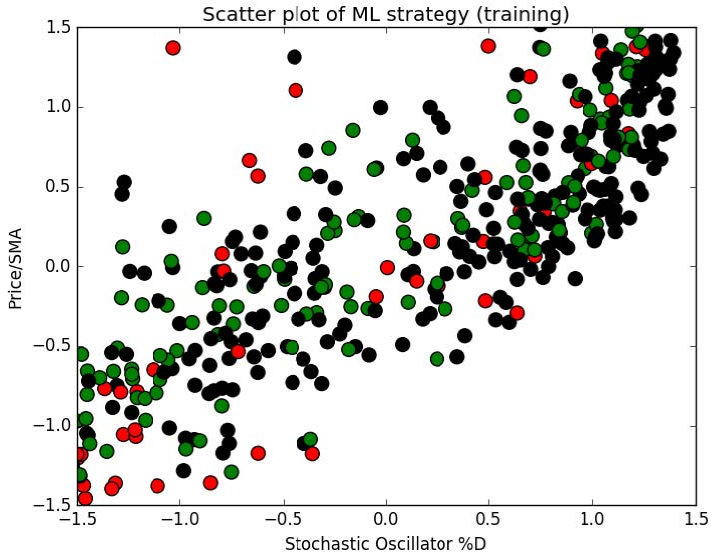
Indicator	'overbought' signal	'oversold' signal
Price/SMA ratio	> 1.05	< 0.95
Stochastic Oscillator %D	> 80	< 20

We will use the scatter plot for pandas, which is easy to use since we already have all the necessary data frames. In the scatter plot, we will mark those days satisfy 'LONG' conditions with green, and 'SHORT' with red. These colors match well with the previous plots. First we can see the scatter plot for the rule-based strategy.

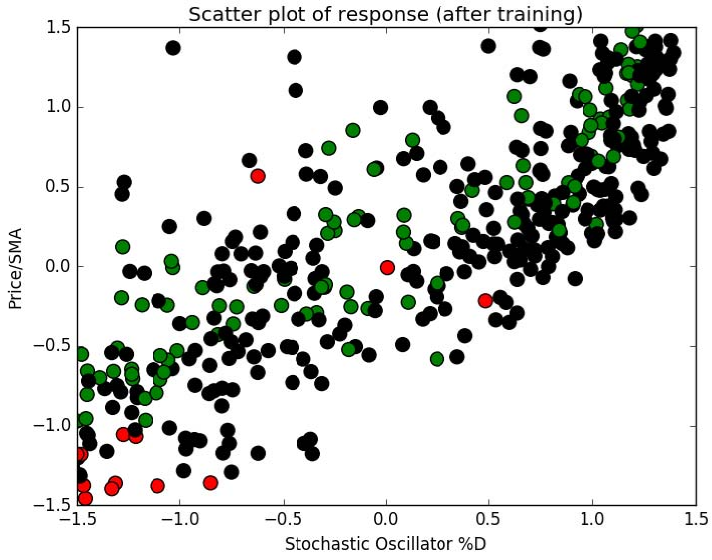


The plot above is very easy to understand if we take a look at the table before. In the center area, we can see all the dots are marked with black. The green dots only occur in the lower left corner, where both Price/SMA ratio and %D indicator are very small. When the values of these two indicators are relatively high, which may trigger a 'overbought' signal, there are lots of red dots there. Also we can see from the above plot the green and red dots are much less than those black dots, which means that we will have only few trading opportunity according to the rule-based policy.

Secondly, we will plot and take a look at the machine learning training result. As we can see from the below plot, all green, black and red dots are quite sparse (unlike these in the rule-based plot). Also we can see there are more green and red dots here.

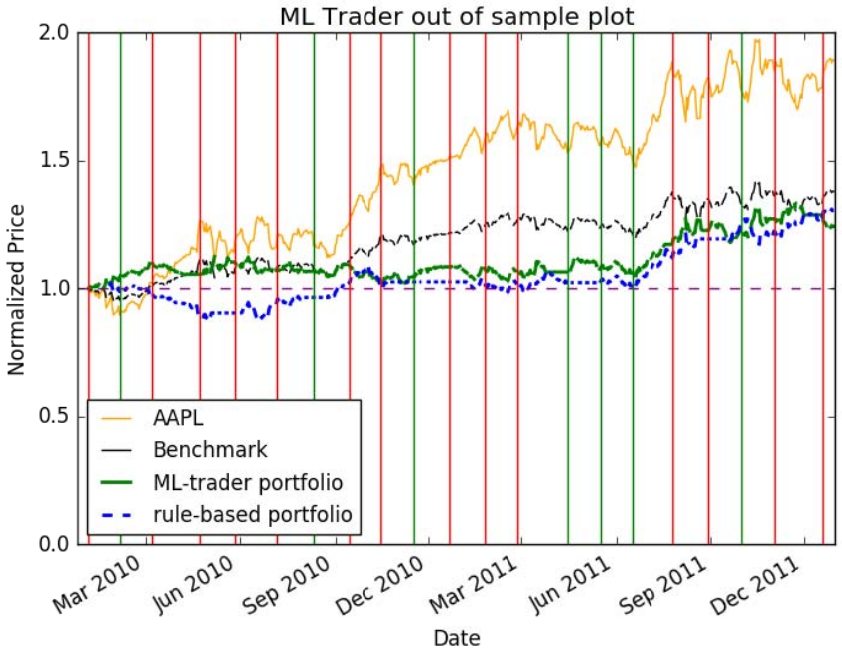


After we successfully train our data, we will process the test or query again on the same period to see whether our learner works. Below is the scatter plot for the response from our learner. Again, we can see that all kinds of colorful dots are very sparse. Then if we take carefully look at the plot and compare it to the training plot, we can see that for those greens dots, the trend and locations of them are quite similar. This is somehow an evidence that our machine learning strategy works well, especially for 'LONG' signals. Also we can see there are more green and red dots compared to the rule-based strategy, which indicates we will have more trading opportunity, and there is a chance we can get more cumulative return.



Part 6. Comparative Analysis

For the last part, we need to use our machine trading learner to do some out of sample test. What we need to do is just change the test set, and then generate a new order file based on the prediction result. Similar as before, we will plot all the benchmark , manual strategy as well as the ML strategy together. Below is the plot along with a summary of in sample and out of sample performance for cumulative return.



	benchmark	rule-based trader	ML trader
In-sample	0.03164	0.2251	0.5554
Out-of-sample	0.3796	0.3053	0.2425

Here I use cumulative return, which I think is the best factor to compare these strategies. As we can see, from in-sample to out-of-sample, the performance of ML trader performs worse, which is common for ML method. The reason could be during the in-sample area, there is usually too much optimization, or the data behaves quite different in the in-sample and out-of sample regions. As for the manual strategy, it performs better, and one possible reason for that is under the test period, the trend is much more simple and consistent (most time the stock price just increases), which will be easier for traditional modeling. Also for the underlying flaw, I think the ML method will be more stable, since it only depends on the data, and when the data is true, then the ML method will be less susceptible.

Summary

From this project we have successfully construct a manual rule-based trading strategy as well as a ML based strategy. We have built several technical indicators and use them for those two methods. By comparing the in-sample and out-of-sample performance, we can see the advantage and disadvantage for these two strategies. Overall it is a quite interesting project and I have learned a lot from it. It is a wonderful and most interesting project~~