Part 1. Indicators

Question 1.1). Is each indicator described in sufficient detail that someone else could reproduce it?

Answer:
For this project I selected five indicators: Price/SMA ratio, Bollinger Band, Money flow index (MFI), MFI of SPY, and momentum.

Price/SMA ratio is the ratio between Price and simple moving average (SMA). SMA is the average price over the last certain period of the stock. Price/SMA is to measure how much current price is relative to the average price over last period.

Bollinger Band value is $\frac{Price-SMA}{2*\sigma}$. This indicator is to measure how much the distance between price and SMA is relative to the moving standard deviation. The value above 1 indicates the stock is over bought, and the value below -1 indicates the stock is over sold.

MFI is an oscillator measuring buy and sell pressure by using both price and volume. First we calculate a typical price = $\frac{High+Low+Close}{3}$, which is the average of high price, the low price and the closing price. Then we calculate money flow = typical price * volume. In a time window, we sum the money flows of all days where typical prices are higher than previous days' typical prices as positive money flow. And we sum the money flows of all days where typical prices are lower than previous days' typical prices as negative money flow. The money ratio is $\frac{Positive\ Money\ Flow}{Negative\ Money\ Flow}$. And then finally we get MFI =100-$\frac{100}{1+money\ ratio}$. When MFI is too high, stock is over bought, and when MFI is too low, stock is over sold.

The fourth indicator I used is MFI of SPY. We calculate MFI using SPY price data. And it reflects the buy and sell pressure of market index.

The last indicator I used is momentum. Momentum is to measure the trend of prices. Given a time window, the momentum is the change of current price compared to past price (momentum[t] = (price[t]/price[t-N])-1).
When momentum is above 0, stock prices have trend to go up. And when momentum is below 0, stock prices have trend to go down.

Question 1.2) Is there a chart for each indicator that properly illustrates its operation?

Answer: Yes. See Price/SMA ratio in fig 1, Bollinger Band value in fig 2, MFI in fig 3, SPY MFI in fig 4 and momentum in fig 5.

Fig.1 Normalized Price/SMA ratio and historical price

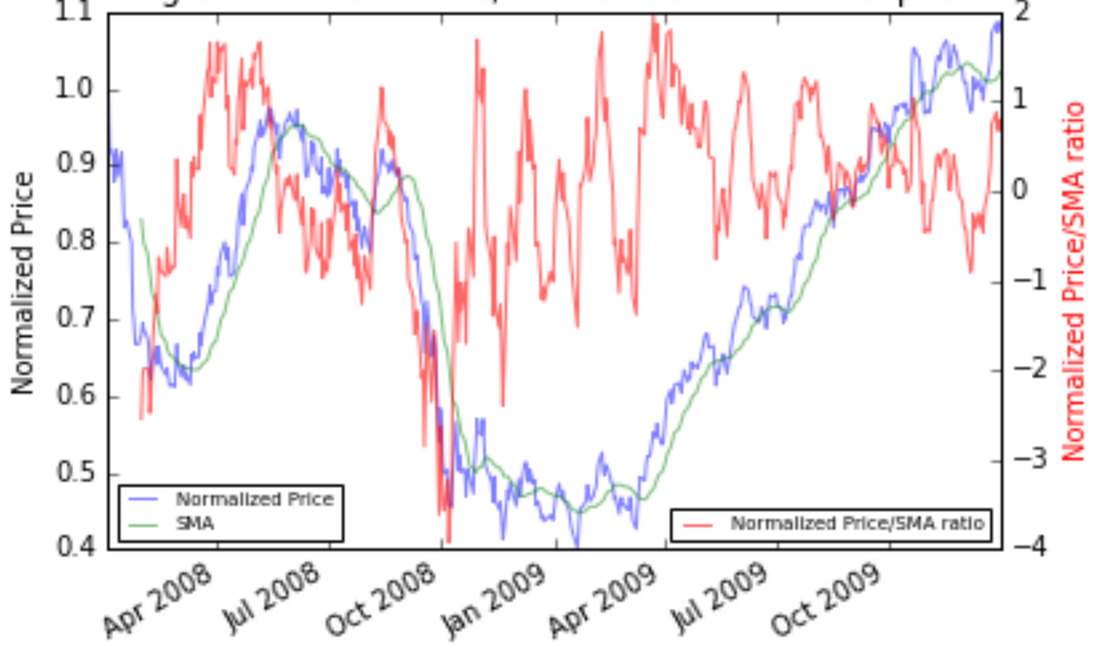


Fig.2 Normalized Bollinger band value and historical price

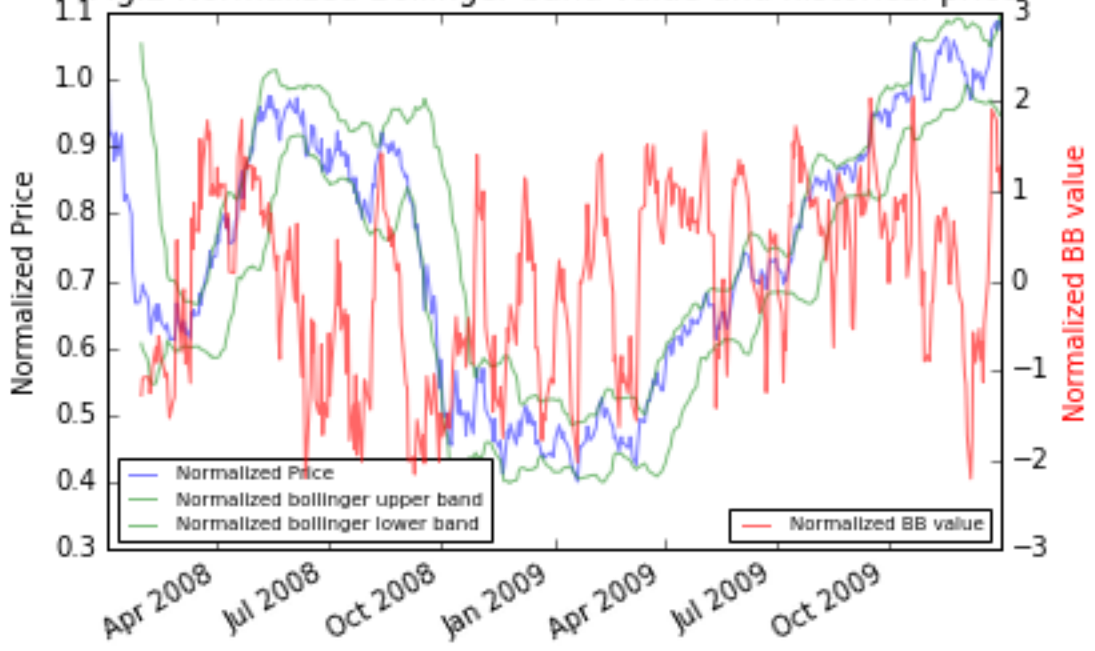


Fig.3 Normalized MFI and historical price

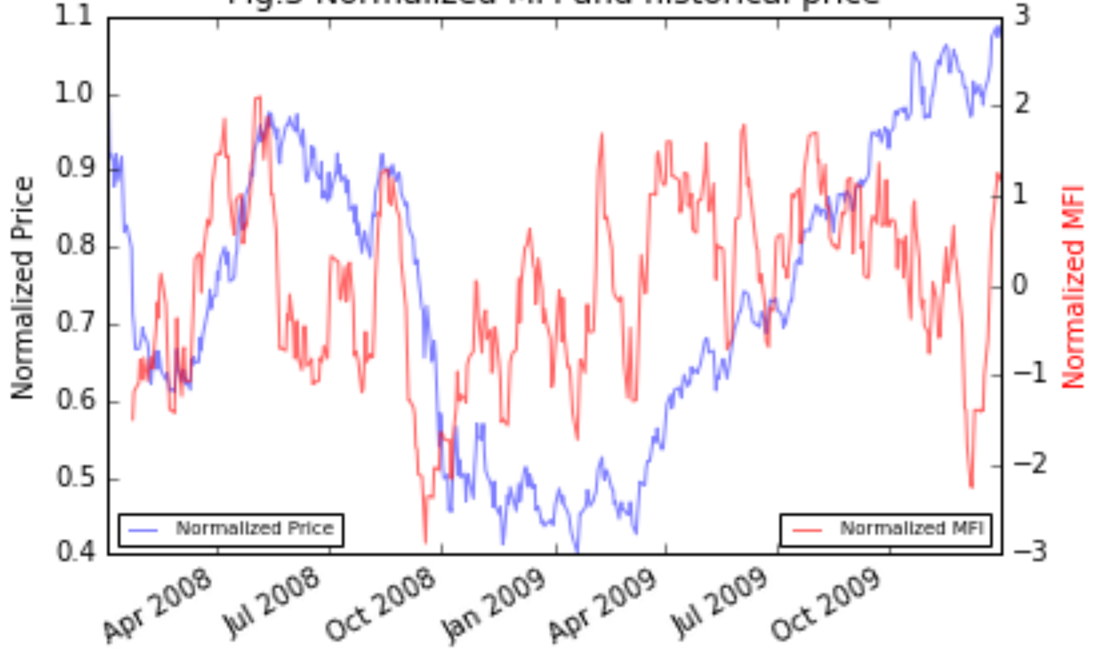
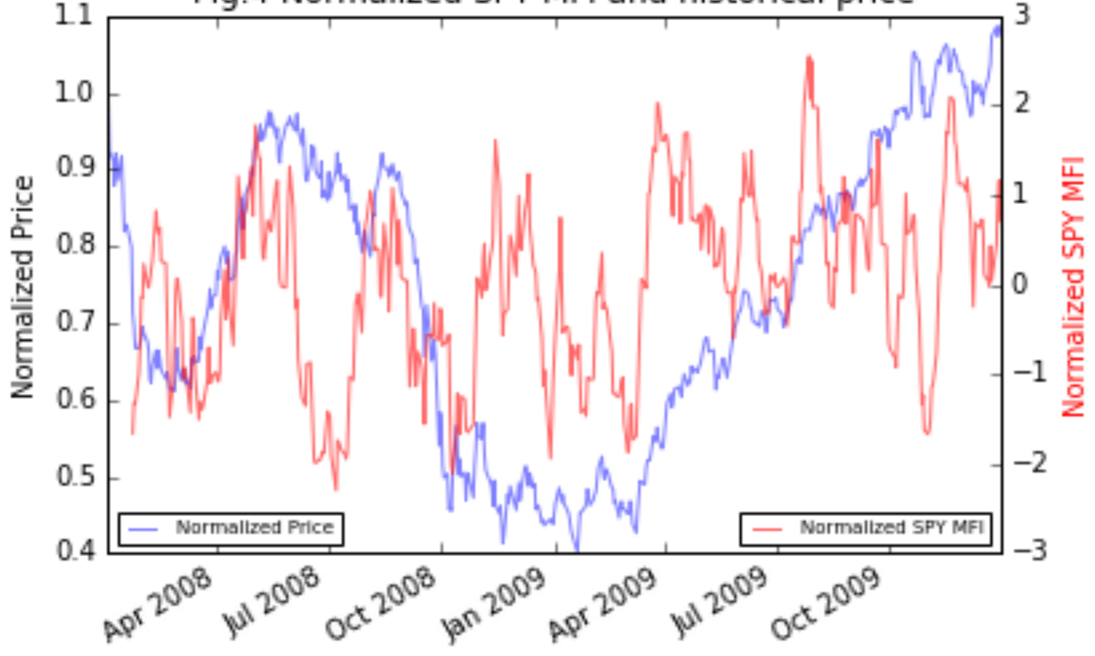
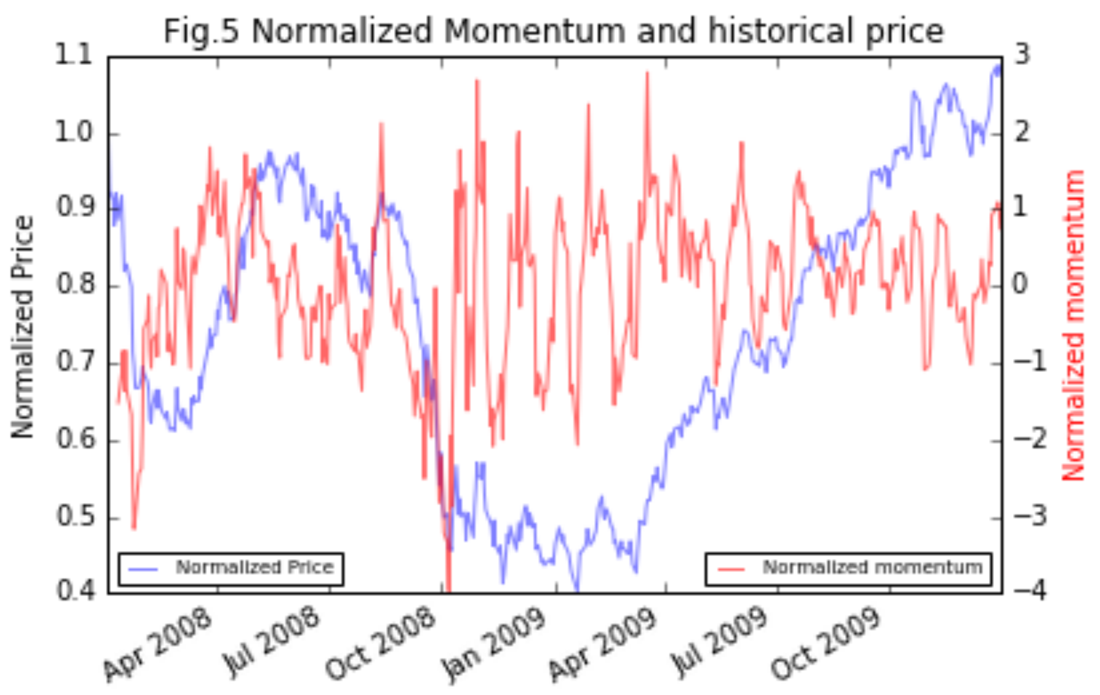


Fig.4 Normalized SPY MFI and historical price





Question 1.3). Is at least one indicator different from those provided by the instructor's code?

Answer: Yes. I used one different, MFI.

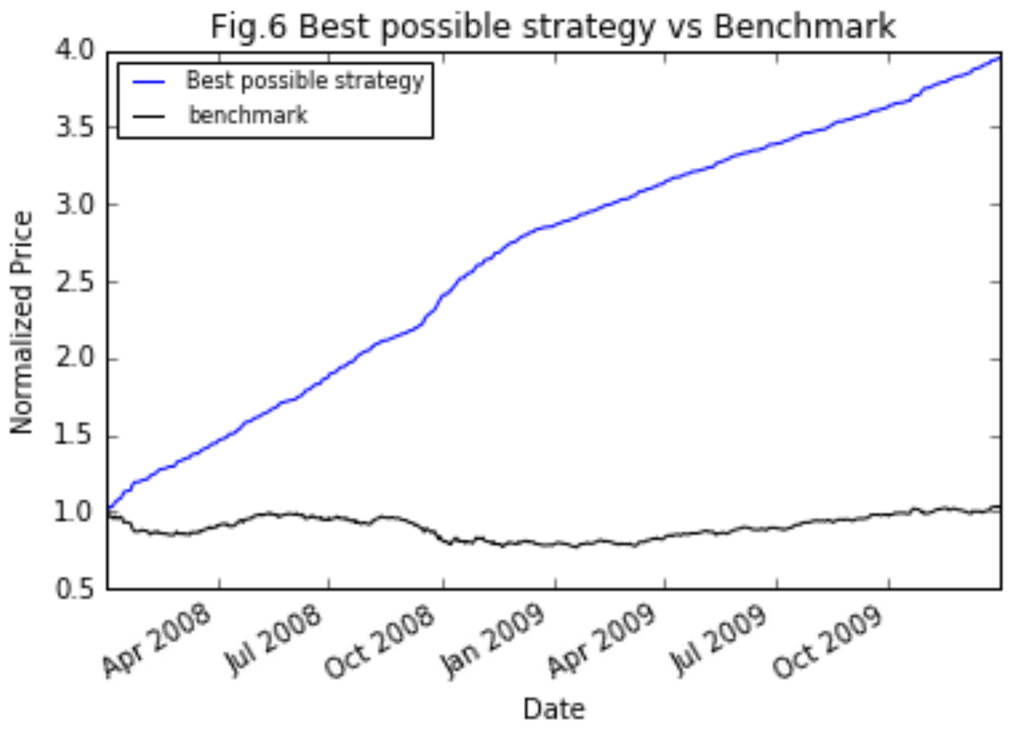
Question 1.4). Does the submitted code indicators.py properly reflect the indicators provided in the report?

Answer: See code, indicators.py. This file contains codes for all calculations, normalizations and plotting of five indicators. Run the file and get the above five figures.

Part 2. Best possible

Question 2.1) Is the chart correct (dates and equity curve)?

Answer: See Fig 6 below.



Question 2.2) Is the reported performance correct within 5%?

Answer: See the report below.

Cumulative Return of Best possible strategy: 2.94948
Cumulative Return of Benchmark : 0.03164

Standard Deviation of daily return of Best possible strategy: 0.00321937085748
Standard Deviation of daily return of Benchmark : 0.00874053752015

Average Daily Return of Best possible strategy: 0.00273420127456
Average Daily Return of Benchmark : 0.000100069116968

Part 3. Manual rule-based trader

Question 3.1) Is the trading strategy described with clarity and in sufficient detail that someone else could reproduce it?

Answer:
When momentum is above 0, but stock is not over bought and market is not over sold, we assign the day a "Buy" signal. Here is the condition:
((norm_momentum >0)& (norm_bbp>0)& (norm_bbp<1.8) & (norm_psr>0) &(norm_mfi>0)&(norm_mfi<1.8)& (norm_spymfi>-1.5)).

When momentum is below 0, stock is over sold but market is not over sold, we assign the day a "Buy" signal 1. Here is the condition:
((norm_ momentum <0)& (norm_bbp<-1.4)&(norm_mfi<-0.9)&(norm_spymfi>-1.5)&
(norm_psr<0))

When momentum is below 0, but stock is not over sold, and market is not over bought, we assign the day a "Sell" signal. Here is the condition:
((norm_momentum <0)& (norm_bbp<0)&(norm_bbp>-1.8)&
(norm_psr<0)&(norm_mfi<0)&(norm_mfi>-0.9)&(norm_spymfi<1.5))

When momentum is above 0, stock is over bought, but market is not over bought, we assign the day a "Sell" signal -1. Here is the condition:
((norm_momentum >0)& (norm_bbp>1.8)& (norm_mfi>0.9)&
(norm_spymfi<1.5)&(norm_psr>0))

For all the other conditions, assign the day 0.

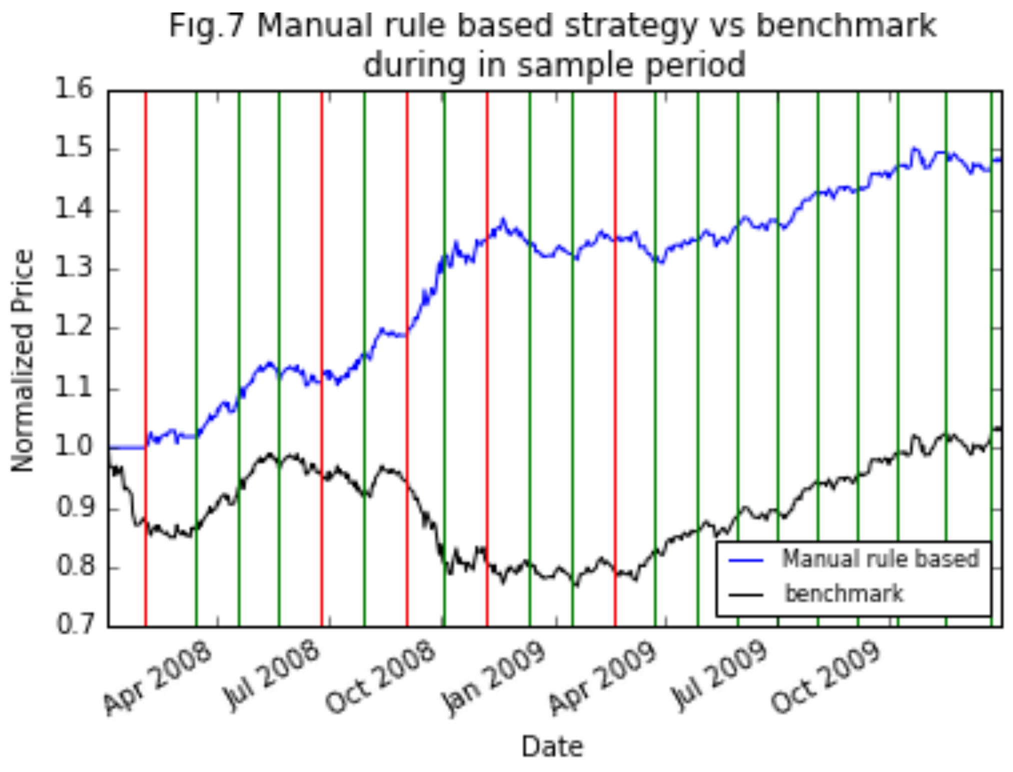
Once we get a signal list of 1, -1, 0, I translate it into an order file by using the holding limit 21 day criteria. Iterate the signal list, when first see a non-zero day, make a corresponding action, either buy or sell 200 shares of AAPL, and ignore signals in the next 21 days, and then liquidate the position. Restart iteration in the next day, and repeat the above process until the last day.

Finally we simulate the portfolio track from the order files by compute_portvals function in marketsim.py

Question 3.2) Does the provided chart include:

- Historic value of benchmark normalized to 1.0 with black line (-5% if not)
- Historic value of portfolio normalized to 1.0 with blue line (-10% if not)
- Are the appropriate date ranges covered? (-5% if not)
- Are vertical lines included to indicate entries (-10% if not)?

Answer: Yes. See Fig. 7 below.



Question 3.3) Does the submitted code rule_based.py properly reflect the strategy provided in the report?

Answer: Yes. See codes in rule_based.py. The file contains codes for normalizing all indicators, building rule_based strategy, creating orders based the strategy, simulating the order file and plotting the portfolios of both benchmark and rule_based strategy.

Question 3.4) Does the manual trading system provide higher cumulative return than the benchmark over the in-sample time period?

Answer: Yes. See either Fig. 7 above or the performance report below. CR of manual trader is much higher than benchmark over in sample period.

Cumulative Return of Manual rule based: 0.48186
Cumulative Return of Benchmark : 0.03164
Standard Deviation of daily return of Manual rule based: 0.00565083919381
Standard Deviation of daily return of Benchmark : 0.00874053752015
Average Daily Return of Manual rule based: 0.000796545491321
Average Daily Return of Benchmark : 0.000100069116968

Part 4. ML-based trader

Question 4.1) Is the ML strategy described with clarity and in sufficient detail that someone else could reproduce it?
Answer:

First translate future 21 day return to Y label for training. Calculate future 21 day returns, and assign them to each day except the last 21 days. Calculate median of positive returns and median of negative returns. Then, I use 0.7* median of positive returns as YBuy, and 0.7*median of negative returns as YSell. If future 21 day return is less than YSell, assign the day "Sell" signal "-1". If future 21 day return is more than YBuy, assign the day "Buy" signal "1". Assign the other days "0".

Then prepare X data for training. Here is the detail. Calculate all five indicators for all possible days and normalize them. Combine the normalized indicators and Y label as a pandas dataframe and drop all NAs. Then split the dataframe to numpy array trainX and trainY.

Change original RTLearner and BagLearner from a regression learner to a classification learner by voting the mode. Then use BagLearner to load evidence of trainX and trainY to build a random forest by bagging.

After query the original trainX by BagLearner, a signal list is generated. Then use the same 21 day criteria as rule based trader to translate the signal list to an order file (See answer to Question 3.1). Then run the order file through market simulator to create a portfolio chart.
Question 4.2) Are modifications/tweaks to the basic decision tree learner fully described?

Answer: Yes
I modified three parameters, YBuy/YSell, bag number, leaf size during in sample period.

For YBuy/YSell, I first summarize future 21 day return in train data, and get median of positive and negative returns. I used the medians as reference and tweaked YBuy/YSell by a ratio. I found when use a ratio 0.7, meaning 0.7*median of positive or negative returns as YBuy or YSell, I get best average Cumulative Return. The results of tweaking ratio were shown below. Since we use random tree to train, the output may be different when re-run the experiment, and you may get slightly different results about tweaking. I also noticed the std is a little big, using ratio 0.5 to 0.8 actually has no significance difference. But I still use the parameter giving the maximum of average return after tweaking. Same strategy also applies to tweaking bags and leaf size.

ratio_to_median	mean of cr	std of cr
0.1	0.589367	0.062651446
0.2	0.610791	0.014180617
0.3	0.51089	0.084497023
0.4	0.568177	0.085571178
0.5	0.615536	0.056159241
0.6	0.631127	0.058598568
0.7	0.642494	0.054221049
0.8	0.622977	0.037623983
0.9	0.590347	0.045123344
1	0.513577	0.090631633
1.1	0.480514	0.107350982

Here I used BagLearner, rather than basic RTLearner. I tweaked number of bags in range of 50 to 400. BagLearner with 200 bags gives best average returns. See the table below.

Number of bags	mean of cr	std of cr
50	0.6611	0.025506202
100	0.659856	0.023858231
200	0.669542	0.015090814
400	0.66305	0.012433443

I also tweaked leaf_size in range of 5 to 9 when BagLearner has 200 bags and 0.7 ratio for YBuy/YSell. Leaf_size 5 gives best average returns. See table below.

leaf_size	mean of cr	std of cr
5	0.68146	0.016816028
6	0.623696	0.05558482
7	0.555124	0.062450658
8	0.44817	0.062207973
9	0.440036	0.094954

In summary, after tweaking above parameters, I decided to use BagLearner with leaf_size 5 and 200 bags, and 0.7 of median positive and negative returns for YBuy/YSell.

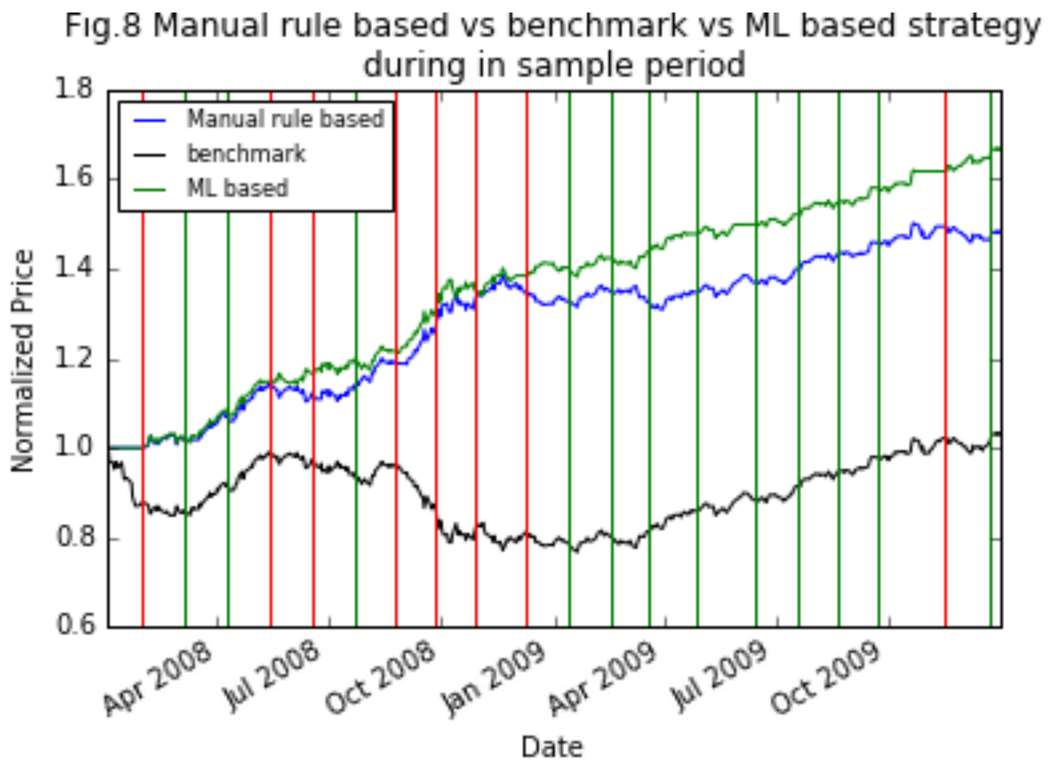
Question 4.3) Does the methodology utilize a classification-based learner?

Answer: Yes. I modified the RTLearner and BagLearner from a regression learner to a classification learner.
In RTLearner, for regression, I used mean of Y values for leaf value, but for classification, I chose mode of Y values for leaf value.
In BagLearner, for regression, I used mean of outputs of all learners as prediction, but for classification, I chose mode of outputs of all learners as prediction.
Details can be seen in code files RTLearner.py and BagLearner.py.

Question 4.4) Does the provided chart include:

Historic value of benchmark normalized to 1.0 with black line
Historic value of rule-based portfolio normalized to 1.0 with blue line
Historic value of ML-based portfolio normalized to 1.0 with green line
Are the appropriate date ranges covered?
Are vertical lines included to indicate entry ?

Answer: Yes. See Fig. 8 below.



Question 4.5) Does the submitted code ML_based.py properly reflect the strategy provided in the report?

Answer: Yes. See the codes in ML_based.py.

Question 4.6) Does the ML trading system provide 1.5x higher cumulative return or than the benchmark over the in-sample time period?

Answer: Yes. See reports below. ML trader makes cumulative return about 20x higher than benchmark.
Cumulative Return of ML based over in sample period: 0.66714
Cumulative Return of Benchmark over in sample period: 0.03164

Part 5. Data visualization

Question 5.1) Is the X data reported in all three charts the same?
Answer:
I chose to plot normalized MFI and normalized momentum in all three charts below. See figs 9-11 below.

Fig.9 Normalized momentum and Normalized MFI in sample by rule based strategy

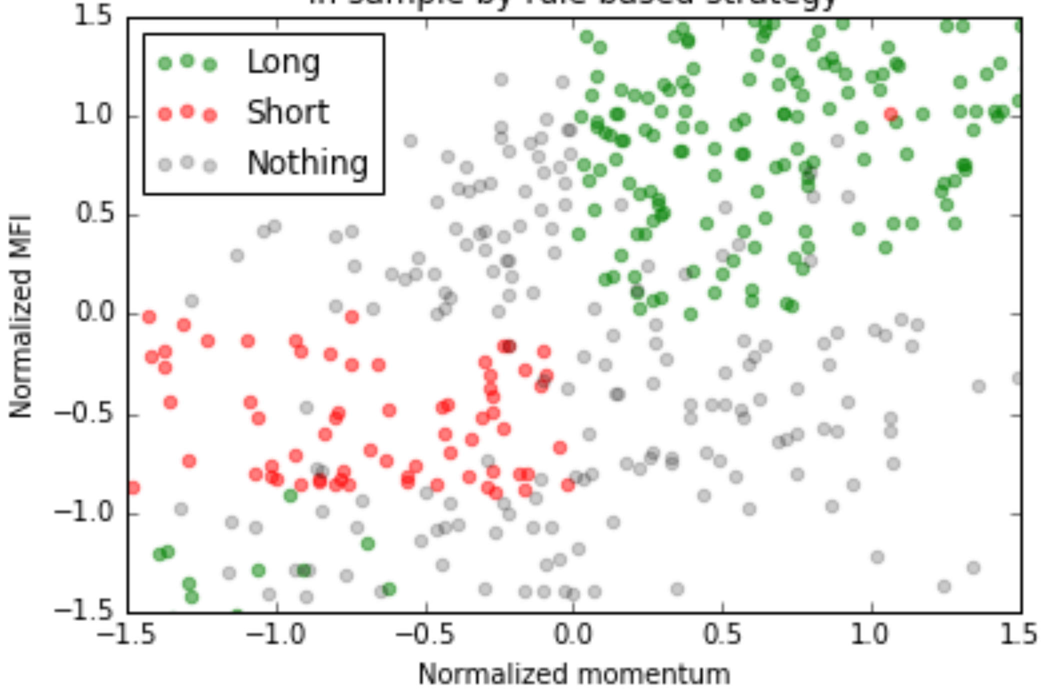
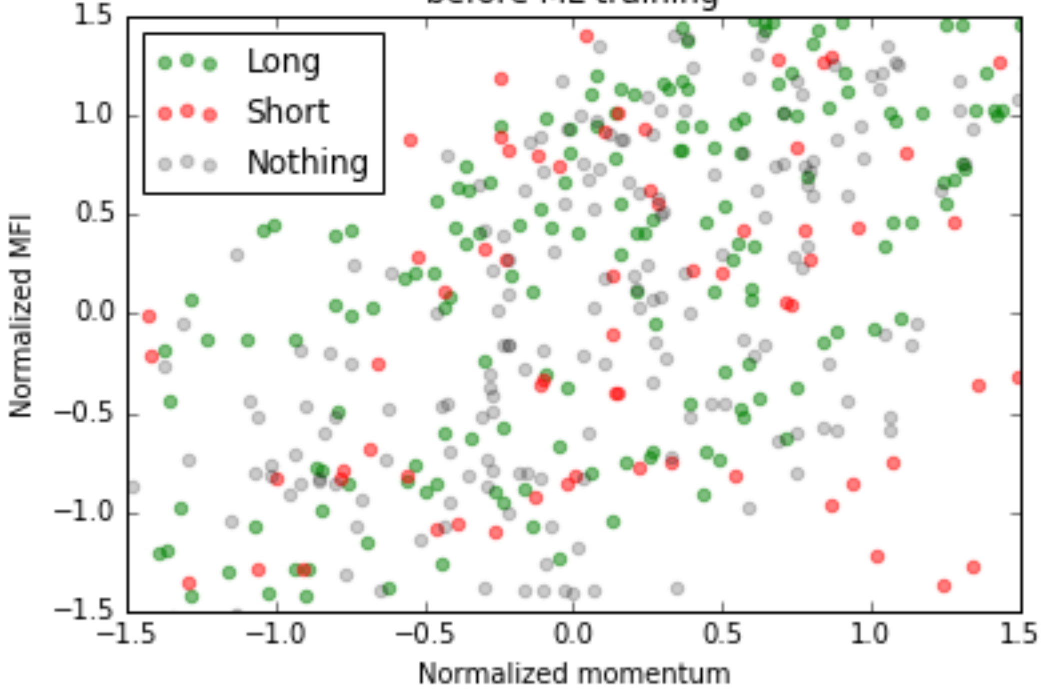
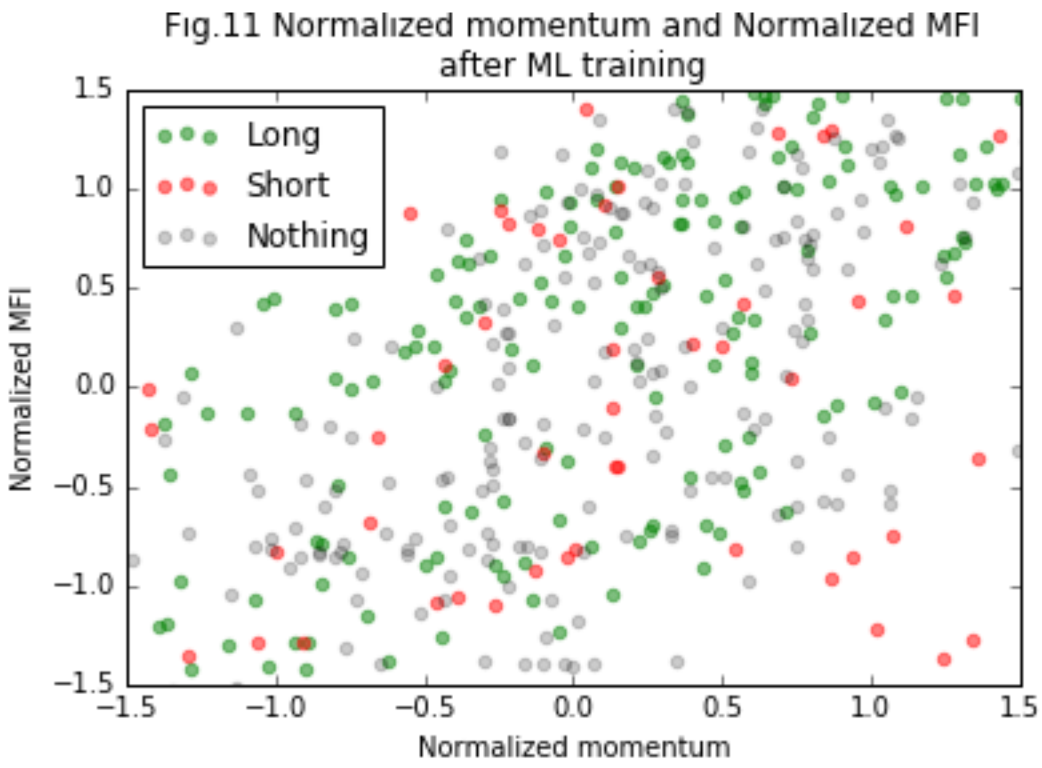


Fig.10 Normalized momentum and Normalized MFI before ML training





Question 5.2) Is the X data standardized?

Answer: Yes. They are all standardized and only Data in range of -1.5 and 1.5 were shown. See figs 9-11 above.

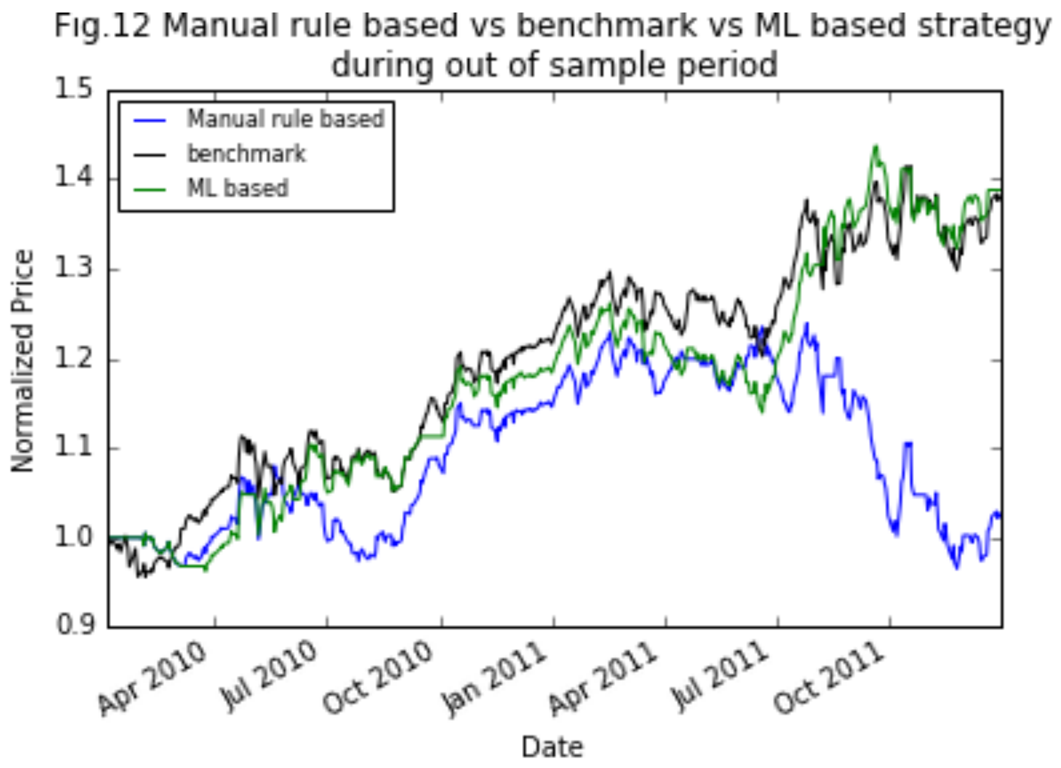
Question 5.3) Is the Y data in the train and query plots similar?

Answer: Yes. See above fig 10 and fig 11. They are very similar. Only several points in total were labeled with different colors.

Part 6. Comparative analysis

Question 6.1) Is the appropriate chart provided?

Answer: Yes. See fig 12 below.



Question 6.2) Is there a table that reports in-sample and out-of-sample data for the baseline (just the stock), rule-based, and ML-based strategies?

Answer: Yes. See the table below. Since random tree is not that stable in performance, I decided to run the traders 20 times and get average and standard deviation of cumulative return and Sharpe ratio. Run comparative_analysis.py and wait for about 4 minutes to get the summaries below. Based on experience in our class, Sharpe ratio is considered as the best factor to use for evaluating performance.

type	mean of Cumulative Return	std of Cumulative Return
Benchmark in sample	0.03164	0
Rule_based in sample	0.48186	1.11E-16
ML in sample	0.627026	0.038361636
Benchmark out of sample	0.38034	1.11E-16
Rule_based out of sample	0.02627	0
ML out of sample	0.3738825	0.119296616

type	mean of Sharpe ratio	std of Sharpe ratio
Benchmark in sample	0.181744885	2.78E-17
Rule_based in sample	2.237679614	4.44E-16
ML in sample	2.969659356	0.171319193
Benchmark out of sample	1.257389713	2.22E-16
Rule_based out of sample	0.16254997	5.55E-17
ML out of sample	1.34665116	0.418237121

Question 6.3) Are differences between the in-sample and out-of-sample performances appropriately explained?

Answer:

From above table, I compared the difference of in sample and out of sample performances, and two strategies (Rule and ML).

First, for both manual rule trader and ML trader, in sample performances are better than out of sample performance. The reason is that we tweak parameters for both strategies over in sample period, and try to get as better as possible in performance. So the models may over fit the in sample period than out of sample period. And it also indicates future is too hard to predict.

Second, ML strategy has better performance than manual rule strategy over both in sample and out of sample period. This makes sense, as market is more complex than we manually summarized. The decision tree strategy finds the model fitting more to the data and giving better predictions than manual rule.

Third, over in sample period, ML trader and rule trader both can beat benchmark. But over out of period, on average, they were not as good as benchmark. Both results are expected. Since over in sample period, we tweaked parameters for both traders, and it is like a cheat to beat benchmark. But over out of sample period, AAPL is in strong bull trend. It is not easy to beat "Buy and hold" for both strategies.

Last question: is one method or the other more or less susceptible to the same underlying flaw? Why or why not?

Answer: Yes, both strategies are susceptible to some underlying flaw. In this project, we didn't do roll forward validation. I only tweaked parameters over the whole training period for both strategies, which may results over fit too much over in sample period. And then in real, these parameters I chose may not make profits. Roll forward validation is necessary in real to make both rule and ML trader successful.

And another flaw is that some of my five indicators may have strong correlations. But only manual rule based strategy is susceptible to this flaw. The random decision tree can overcome this issue, since it selects features randomly to split tree.