Report

Part 1: Technical Indicators

- Overall

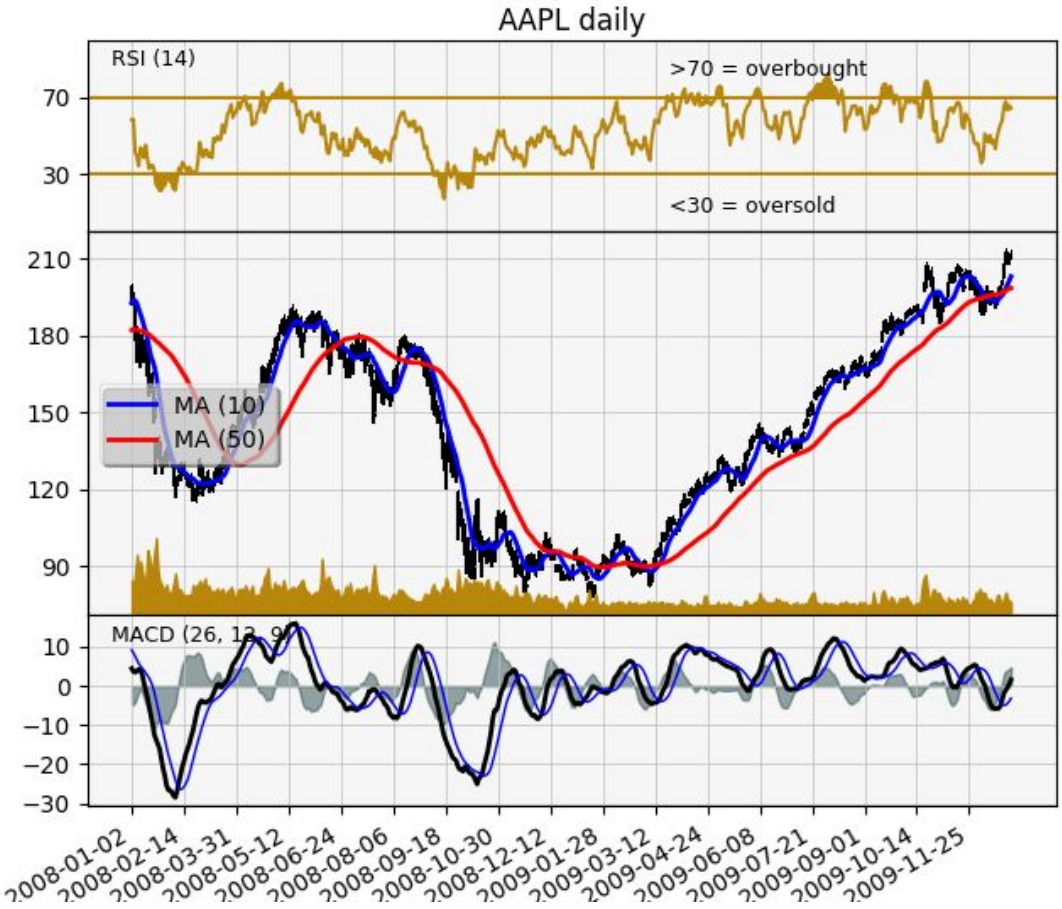


Fig 1. APPL Daily Chart

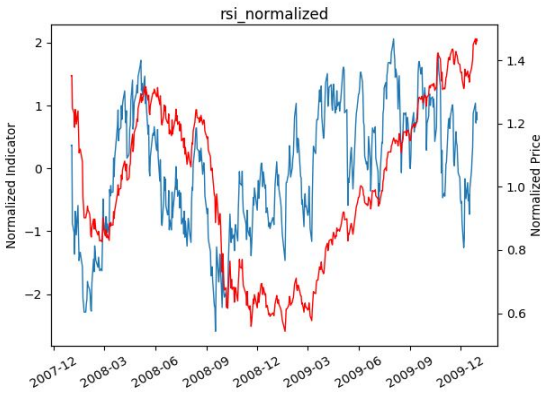
Fig 1. Is a high level overall chart of APPL stock, including Price (candlestick chart), Volume, MA 10/50, RSI, and MACD.

For RSI, when RSI value larger than 70, normally it is considered as overbought, may consider to sell, and when it is smaller than 30, should buy since it is oversold.

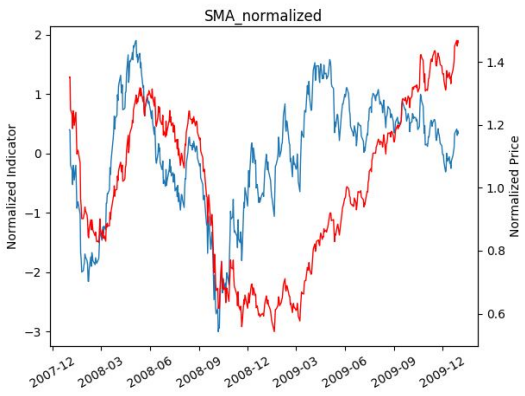
For MA, we can use different period, MA10 for 10 days and MA 50 for 50 days, and it is usually considered that it is a buy point when MA10 crosses MA50 from bottom, and a sell point when MA10 crosses MA50 from top. SMA and MACD (Moving average convergence divergence) are used to capture feature.

Bollinger Band and momentum from the class are also evaluated in this project.

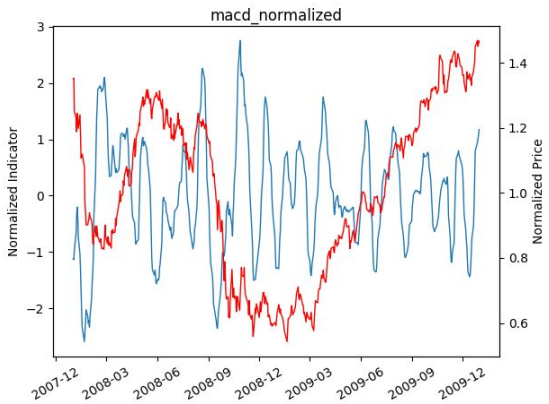
- Normalized Indicators



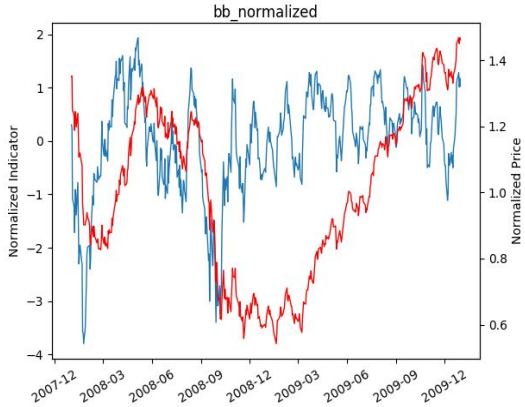
(a)



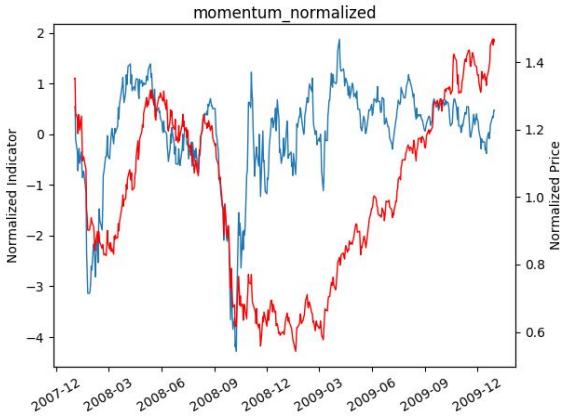
(b)



(c)



(d)



(e)

Fig 2. Normalized Indicators vs Normalized Prices

I have chosen 5 indicators:

1. RSI_normalized

Normalized Relative Strength Index is a momentum oscillator that measures the speed and change of price movements. Just need to define a window=n, and take both positive and negative price change average in this window defined as up and down baselines. Init the rsi list with first window then slide the window along the trading day axis.Collect gain if price change is positive and loss if price change is negative, then compute new up to down ratio as rs based on the accumulated up and down average and rsi is computed as 100. - 100./(1. + rs)

2. SMA_normalized

Normalized Simple Moving Average. Loop into the price to get average up to the looped point from the past n days to get MA. Then use *SMA% = (close - MA) / MA* to get SMA.

3. MACD_normalized

Normalized Moving Average Convergence Divergence. Since I already implemented MA, just use a fast and slow exponential MA and find the length difference.
U'momentum_normalized':

4. Momentum_normalized

Normalized momentum. Take two price change percentage based on two days with apart of N days when days are moving along the day axis. Here N=20.

5. Bollinger _normalized

Normalized Bollinger band indicated, which is calculated by (prices - SMA)/prices_stdev

Part 2: Best Possible Strategy

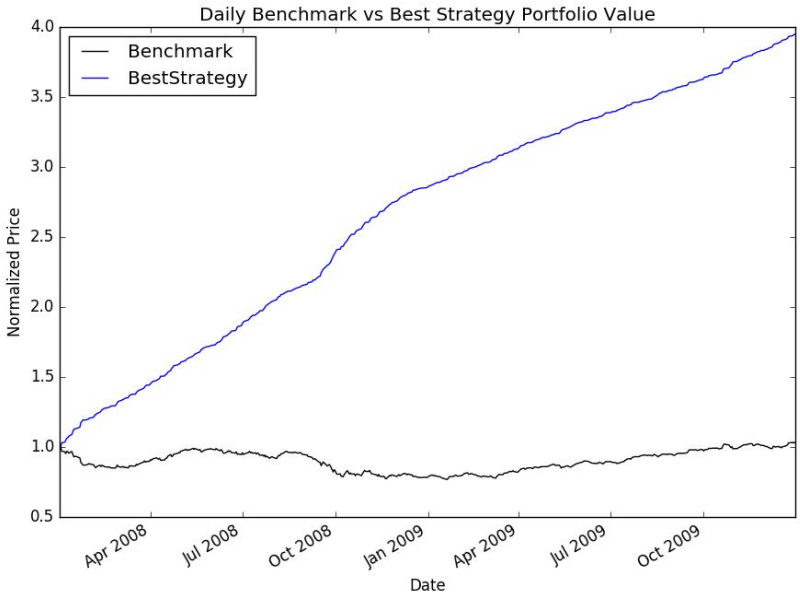
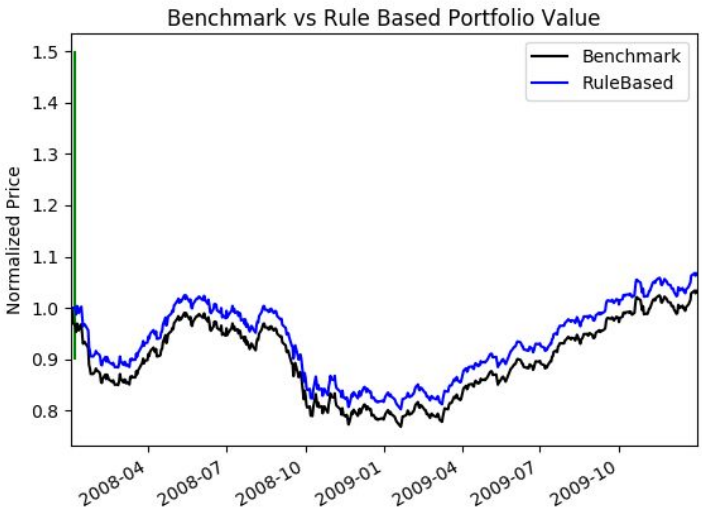


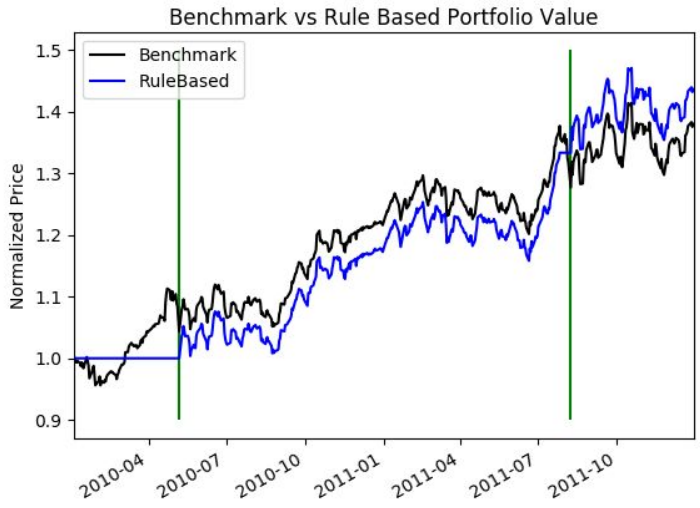
Fig 3. Benchmark vs Best Strategy

	Cumulative return	Stdev of daily returns	Mean of daily returns
Benchmark	0.03164	0.00874	0.0001
Best strategy	2.94948	0.003219	0.002734

Part 3: Manual Rule-Based Trader



(a)



(b)

Fig 4. Rule Based vs Benchmark

My rule based strategy is based on three indicators (RSI, MACD, Bollinger Band), and for each indicator I set different values for long sell/short buy, and short recover/long buy, to triggle the four operations. This means I didn't do short simultaneously when sell, short usually is harder to triggle than sell, only when the indicator goes even higher. Similar to Short recover

and Buy. The reason I am doing so is that because usually in man-controlled trading, nobody is doing short immediately after sell, which is pretty risky. Please refer to the code for the detailed triggle points.

The logic is:

- BUY: when all three indicators are smaller than the buy point values, and no holdings
- SELL: when all three indicators are larger than the sell point values, and hold > 21 days
- SHORT: when all three indicators are larger than the short point values, which is usually larger than the sell point values, and no holdings
- RECOVER: when all three indicators are smaller than the recover point values, which is usually larger than the buy point values, and hold > 21 days

Fig 4. shows the rule based strategy vs benchmark portfolio. My rule based strategy outperforms the benchmark in both of the two time period.

Part 4: ML Trader

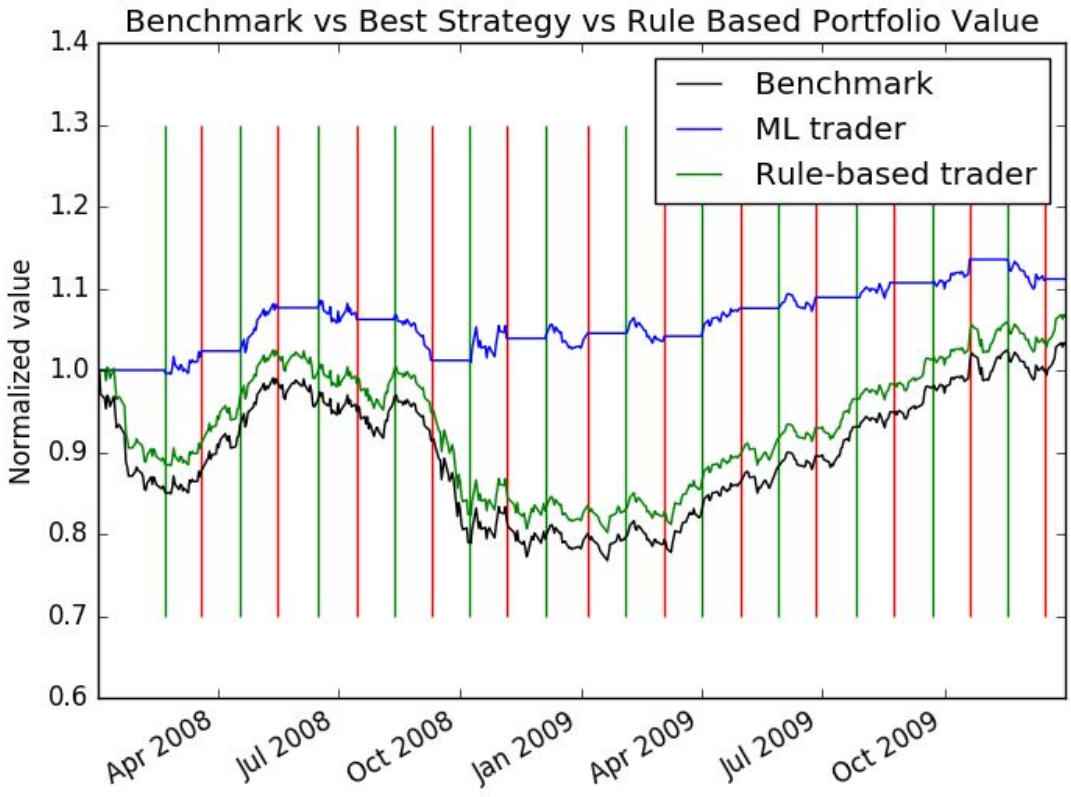
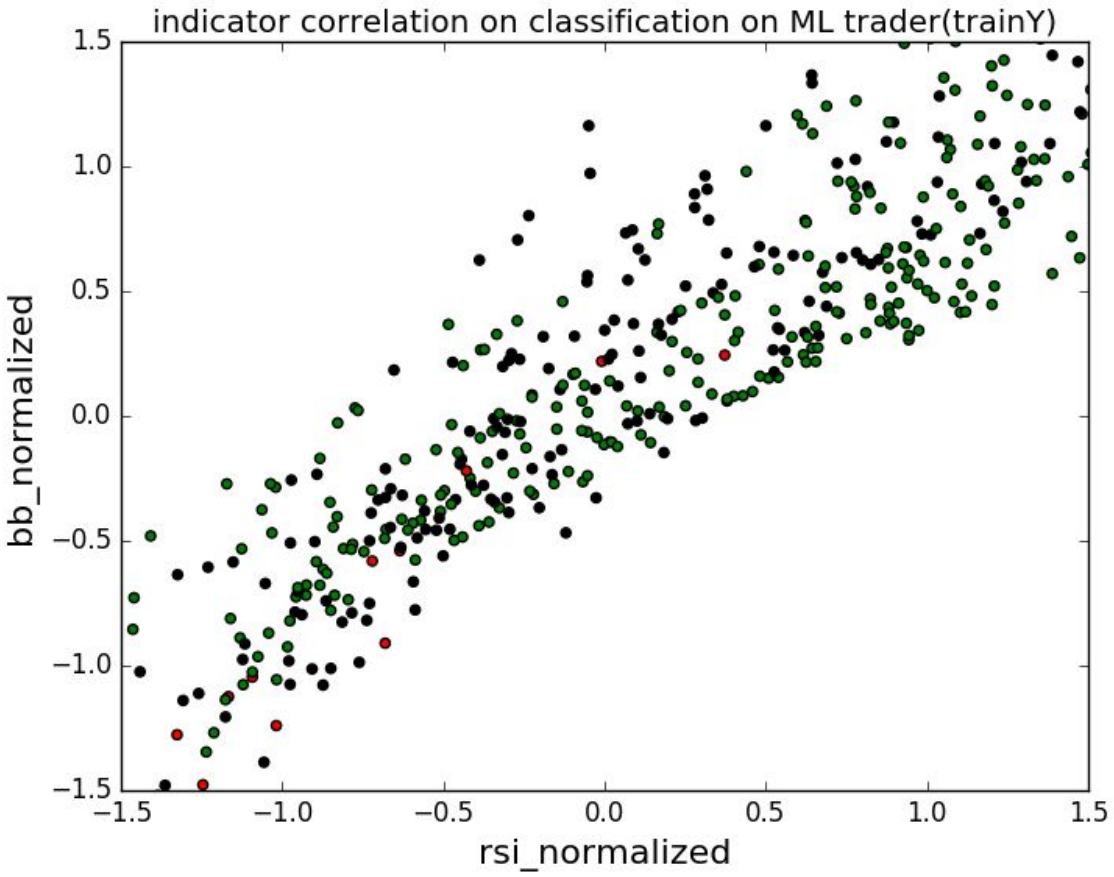


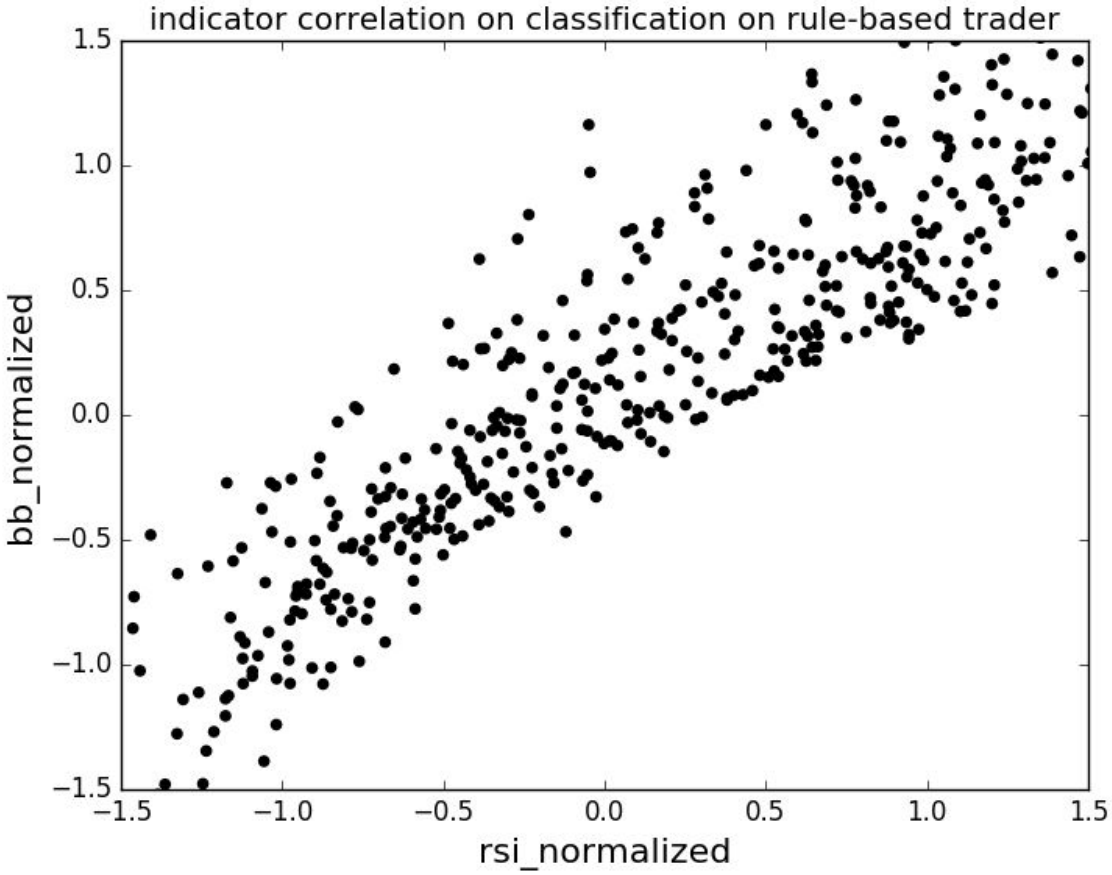
Fig. 5 Benchmark vs Rule-Based trader vs ML trader for learning period

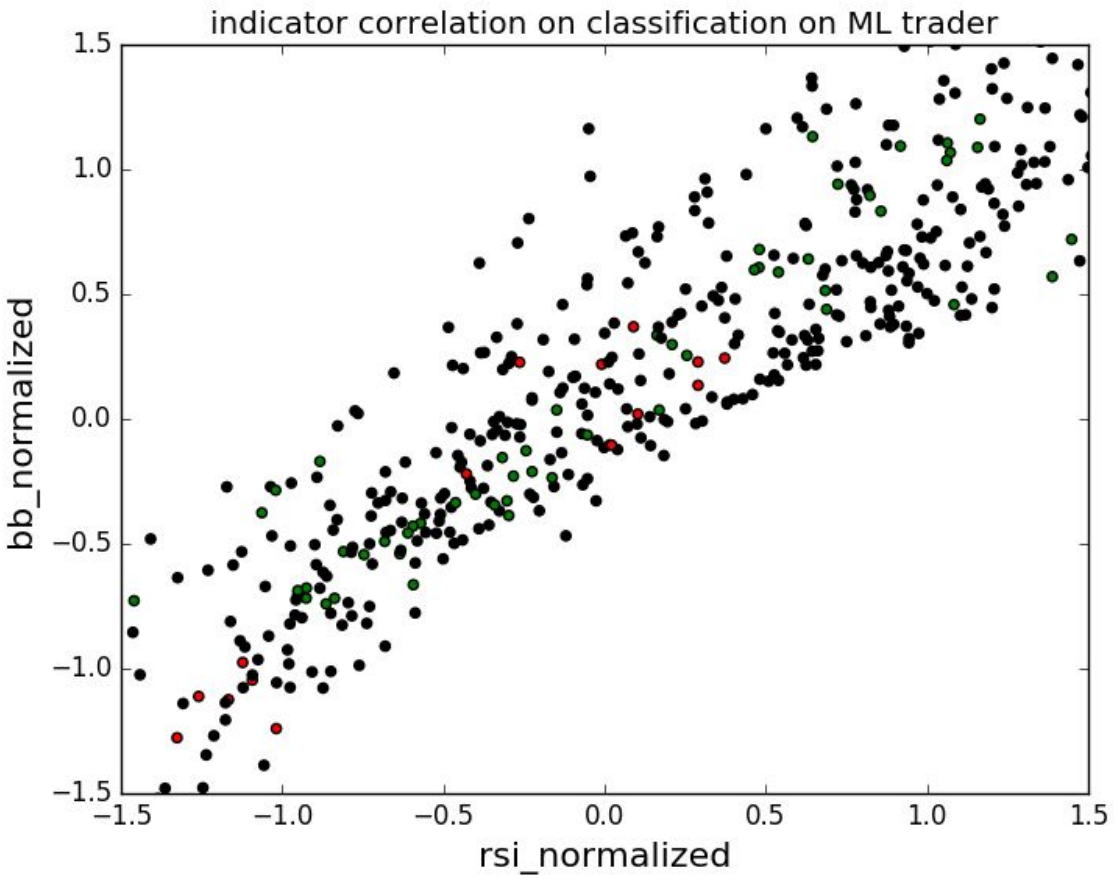
First I modified my RT learner from regression to classification by taking mode instead of mean at each split. Then I created a training and testing file generator, which generates a set of

training and test data with aligned 5 indicators with YSell and Ybuy labels. The Y labels are created based on 21 day's return and preset YSELL and YBUY values. Then my ML_based calls train and test on the training and test files I generated previously, then output the classified test files in the format of order file. Then the market simulator takes the order file and produce the portfolio. In order to tweak the best performance of ML trader, I wrote a loop with loop to figure out the best combination of YSELL, YBUY, leaf_size and the bag size.
In terms of valid order file, I implemented the 21 days trading rule based on the original classified order file. I force any open position to be open for exactly 21 days, so if I sell on dayx, I buy on dayx+21 and verse vesa, in this way, a lot of BUY and SELL actions are removed from the original order file and the new order file is delivered to market simulator.

Part 5: Visualization of data




indicator correlation on classification on rule-based trader
bb_normalized
rsi_normalized
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5


indicator correlation on classification on ML trader
bb_normalized
rsi_normalized
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5

Part 6: Comparative Analysis

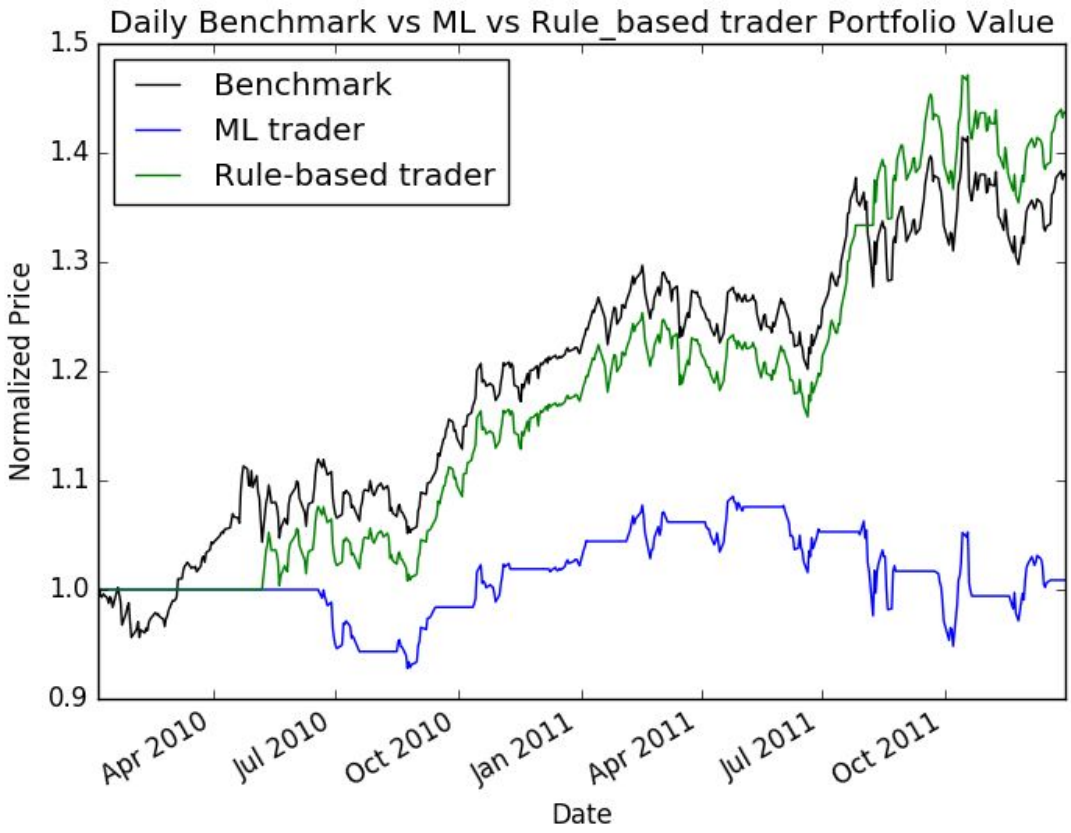


Fig. 9 Benchmark vs Rule-Based trader vs ML trader for comparable period

	stock	rule-based	ML
Cumulative Return	0.38034	0.43682	0.00862
Standard Deviation	0.00855	0.007648	0.00716
Average Daily Return	0.0006775	0.000749	4.26867190701e-05
Final Portfolio Value	138034.0	143682.0	100862.0

Table2, out-sample performance

My ML trader cannot be tuned to 1.5x in-sample anyway with the indicators I picked, since I run 4-loops parameter optimization to explore the best model in the whole space. So, I can confirm the indicators that I picked are not appropriate but I tried price/SMA and volatility as well, they are not helping my model. So based on in sample and out sample result, rule-based trader is always the best, followed by benchmark, then the ML trader. But both my ML trader and rule-based trader are gaining profit at the end.

	stock	rule-based	ML
Cumulative Return	0.03164	0.0659	0.03574
Standard Deviation	0.00874	0.00831	0.00523
Average Daily Return	0.0001	0.000161	8.32782060793e-05
Final Portfolio Value	103164.0	106590.0	103574.0

Table3, in-sample performance

Table 2 and 3 are some stats comparison, I found they are all important in the comparison. So cumulative return can tell absolute gain or loss for the 3 methods, and rule-based trader is consistently deliver the best cum return. While std is way to tell how variable is the trader, in this case, ML trader delivers a slow and flat growth instead of steep increase or jump, I guess this is the only merit of the ML trader. If you look at the rule-based plot, it closely follows benchmark, and just ensure a positive gain over benchmark is my rule, hence, the drop is bigger compared to the ML trader. At certain point of day during a year, it has some very low balance. Next is the averaged daily return, it's important because it tells me despite of huge gain one day or huge loss one day, what can I earn in average if I use this trader. And rule-based trader wins becoz it's above benchmark most of the time even only a bit and the majority of the time, it's gaining instead of losing.
So overall rule-based trader is the best and steady.
If I compare in-sample and out-sample performance, my in-sample is better. For ML, it's expected, because in-sample has some degrees of over-fit all the time. For rule-based, if the stock price is very different than in-sample, I cannot summarize it because future is invisible. So I guess my rule-based trader is just based on the shape of the in-sample year. My ML might have some flaw related to the indicators that I pick so that it consistently deliver poor performance. But unlike other people's result, a great fit in-sample, but very poor performance out-sample, my ML has less over-fit which could be used by conservative users.