MC3 – P3

Manqing Mao (mmao33)

Part 1: Technical Indicators

In part 1, I have chosen three technical indicators: simple moving average (SMA), exponential moving average (EMA) and Bollinger Band (BB). Note that I also choose the window size of 20 days to generate three plots for three indicators.

1.1 Simple Moving Average (SMA)

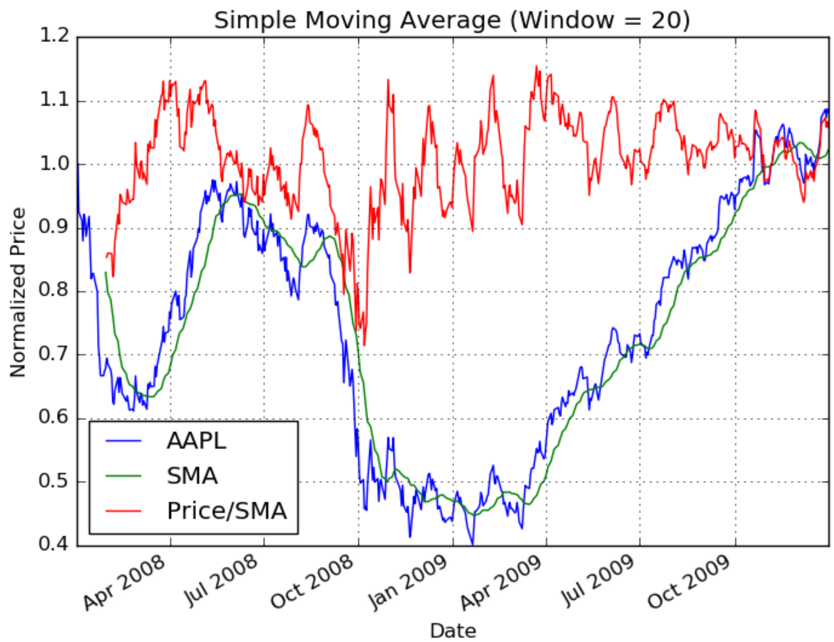

Fig. 1.1 SMA for AAPL with window size of 20 days

A simple moving average (SMA) is an arithmetic moving average is the average closing price for a certain stock within a time window. It is calculated by adding the closing price of a stock ('AAPL' here) for a number of time periods (= 20days) and then dividing this total by the number of time periods (= 20 days).

1.2 Exponential Moving Average (EMA)

An exponential moving average (EMA) is very similar to the SMA, which is mentioned above. EMA is an average price calculation over a specific time period that puts more weight on the most recent price data causing it to react faster to price change. In the code, I used pandas library: pandas.ewma() to calculate the EMA.

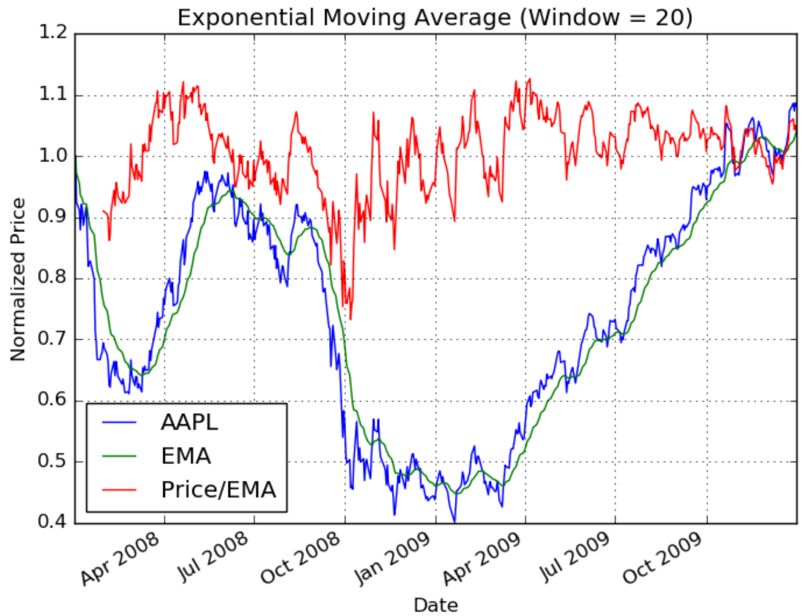

Fig. 1.2 EMA for AAPL with window size of 20 days

1.3 Bollinger Band (BB)

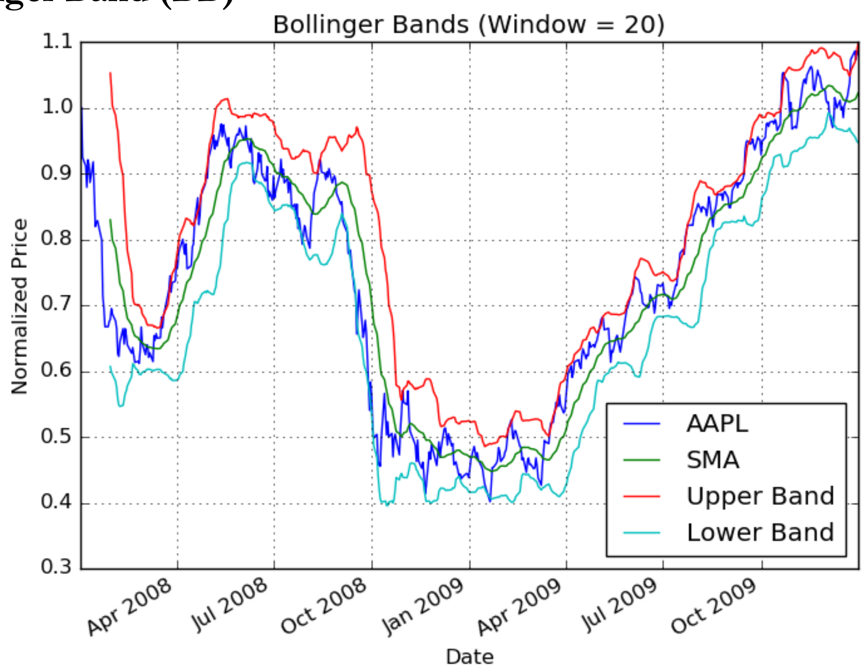

Fig. 1.3 BB for AAPL with window size of 20 days

Bollinger Bands consist of:

(1) SMA: an N-period moving average (N = 20 days);

(2) Upper Band: an upper band at K times an N-period standard deviation above the moving average (MA + Kσ), (K = 2);

(3) Lower Band: a lower band at K times an N-period standard deviation below the moving average (MA − Kσ), (K = 2).

Part 2: Best Possible Strategy

In part 2, we can get best possible strategy if we can peek into future. Then, we can get the upper bound on performance. The strategy is that we can long the stock for 200 shares if tomorrow's return is > 1 while we short the stock for 200 shares if tomorrow's return is < 1.

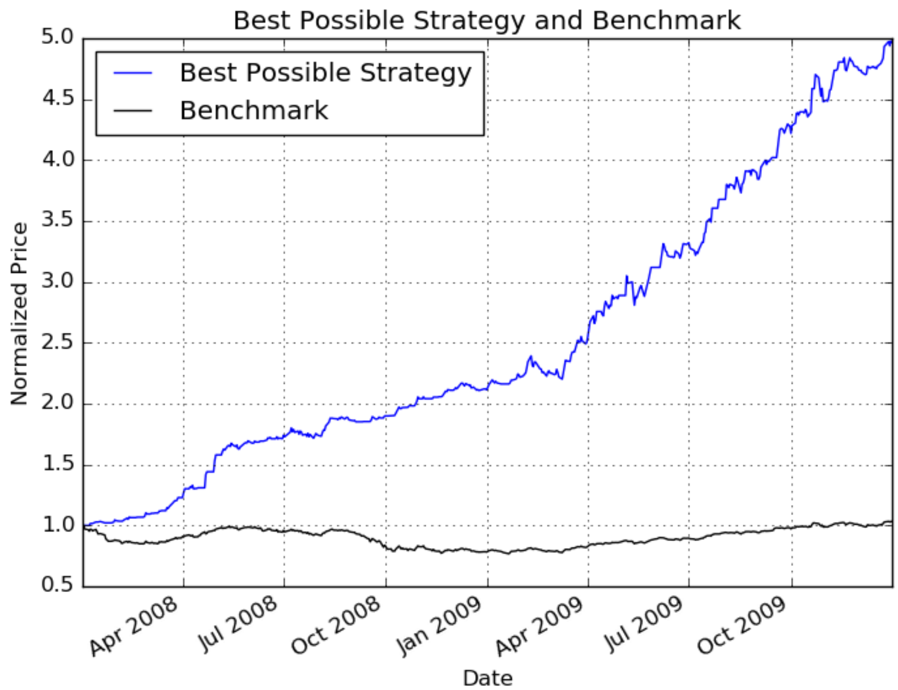

Fig. 2.1 Best possible strategy v.s. benchmark price

We get the final value of the benchmarks is 103164 while the upper bound (the best strategy) is 496648. The other parameters like cumulative return, volatility and mean of daily return for best strategy and benchmark in shown in Fig. 2.2.

```
Final Value for Best Strategy: 496648.0
Final Value for Benchmark: 103164.0
Cumulative Return for Best Strategy: 3.96648
Cumulative Return for Benchmark: 0.03164
Volatility for Best Strategy: 0.0124930624083
Volatility for Benchmark: 0.00873186204653
Mean of Daily Return for Best Strategy: 0.00326188919218
Mean of Daily Return for Benchmark: 0.000100069116968
```

Fig. 2.2 Other parameters for best possible strategy v.s. benchmark price

Part 3: Manual Rule-Based Trader

In part 3, I have developed a rule-based method to generate orders book. The rule-based method is developed by:

Case I (BUY): SMA < 0.95 and bbp < 0
Case II (SELL): SMA > 1.05 and bbp > 1
Case III (HOLD).

Where bbp is calculated by using (price - lower band)/(upper band – lower band). To be more specific, when bbp < 0, the price is below the lower band so that it is the chance to long the stock. Similarly, when bbp > 1, the price is above the upper band so that it is the chance to short the stock.

Another technical indicator is SMA. When SMA < 0.95, the price is quite low so that it is the chance to long the stock. Similarly, when SMA > 1.05, the price is higher than the average so that it is the chance to short the stock.

In this way, we could develop the rule-based method based on these two technical indicators to generate the orders book. Then, the orders book could be fed into market simulator to draw the figure.

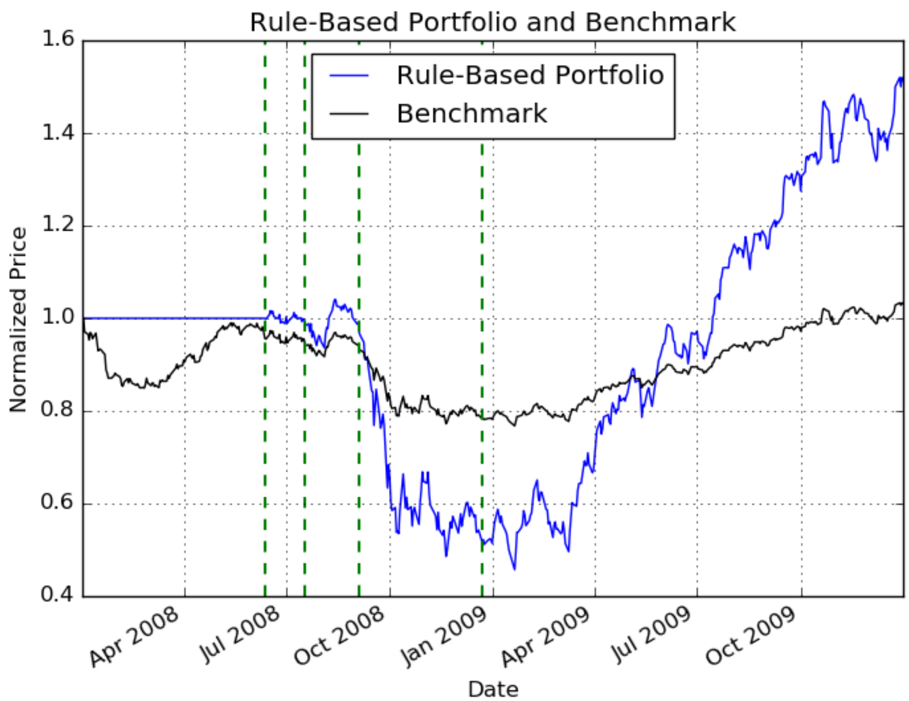

Fig. 3.1 Rule-based Porfolio v.s. benchmark price

Part 4: ML Trader

In part 4, a ML-based trader is developed to optimize the trading actions. Several steps are involved to build the ML trader:

Firstly, the training data have to be generated to feed into Random Tree learner to train the model. The training data is consisting of three technical indicators (X1, X2, and X3) values and Y. Y here is the trading action:

Case I (LONG): ret > YBUY → Y = 1
Case II (SHORT): ret < YSELL → Y = -1
Case III (HOLD): → Y = 0

Where **ret** reflects the 21-day change in price. ret = (price[t+21]/price[t]) – 1.0. YBUY is set to be 0.05 and YSELL is set to be -0.05. Then, we can generate the training data including X1, X2, X3 and Y and save it as `data-training.csv.`

Secondly, I use the RT Learner and Bag Learner to train to data in `data-training.csv` and get the model. The leaf size is set to be 5 and the number of bag is set to be 2. After training the data, we get predictions for Y during in sample period. Then, we can get orders book named `orders-ML.csv` during in sample period.

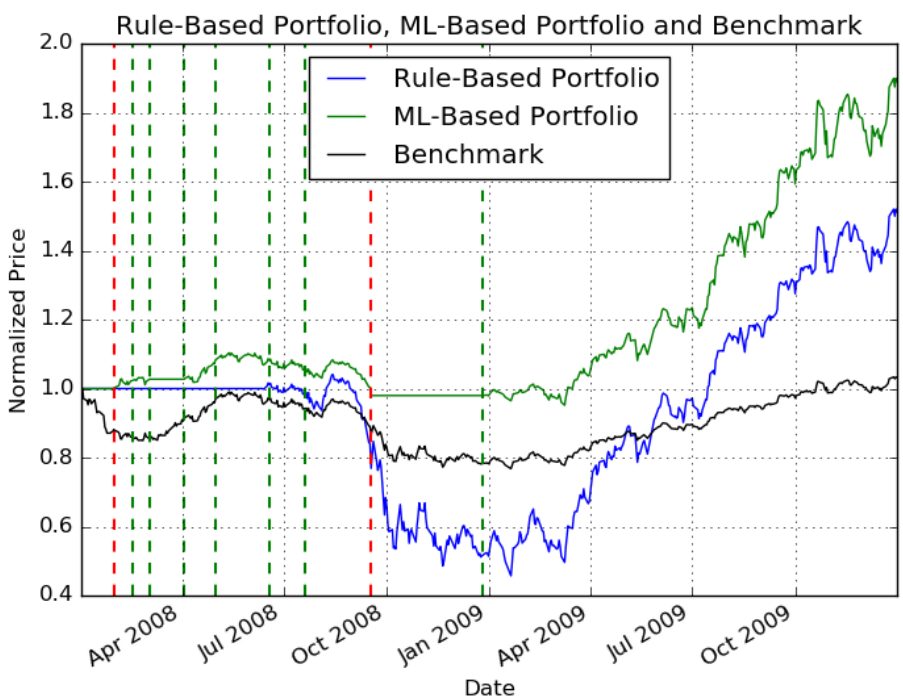

Fig. 4.1 Rule-based Porfolio, ML-based Potfilio v.s. benchmark price during in sample period

Next, the market simulator is used to get the table of values according to the orders book. Both orders book of Rule-based and ML-based method are fed into the market simulator to generate the following figure. Benchmark is the baseline.

From the Fig. 4.1, we can see that the ML-based portfolio shows the highest return while the benchmark has the lowest return at the end of day. ML-based strategy outperforms the manual rule-based strategy.

<u>Part 5: Visualization of Data</u>

In part 5, I have chosen two technical indicators of SMA and BBP. I used BBP here instead of BB to normalize the technical indicators. For three following figures, the X axis is the normalized SMA and Y axis is the BBP.

For Fig 5.1, we show the 'LONG' points and 'SHORT' points based on rule-based strategy. For Fig. 5.2/5.3, we show the 'LONG' points and 'SHORT' points based on ML-based strategy before/after training.

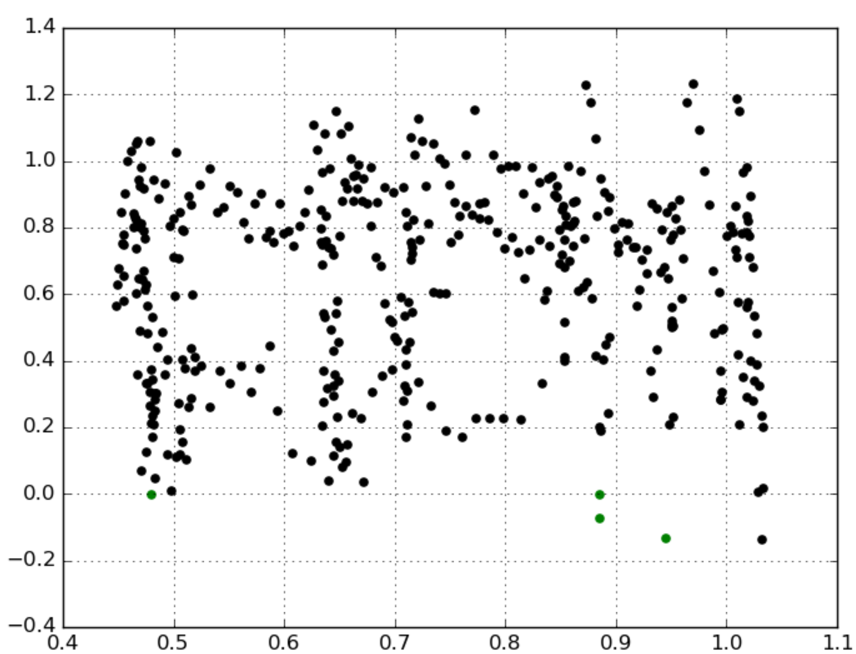

Fig. 5.1 Normalized SMA vs. BBP for the rule-based strategy. the X axis is the normalized SMA and Y axis is the BBP.

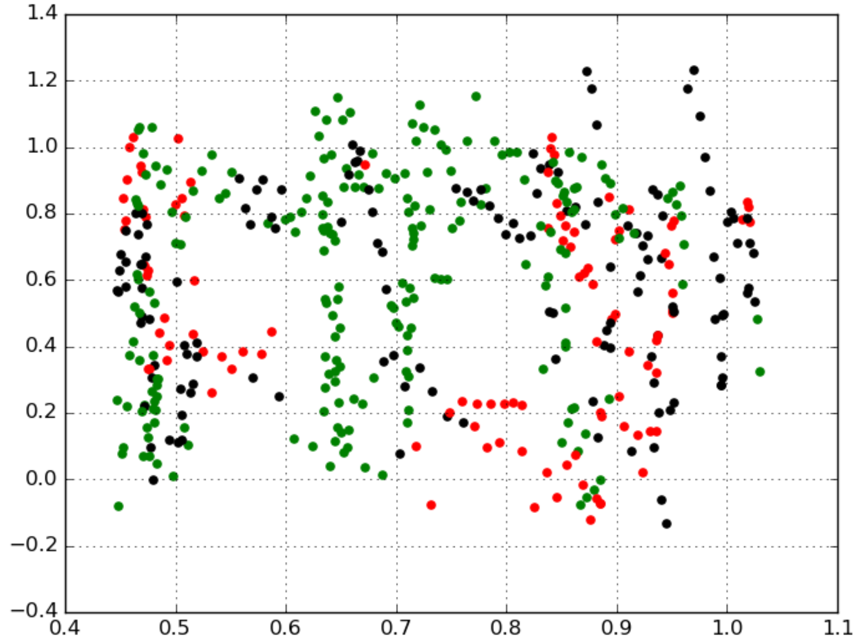

Fig. 5.2 Normalized SMA vs. BBP for the ML-based strategy before training. the X axis is the normalized SMA and Y axis is the BBP.

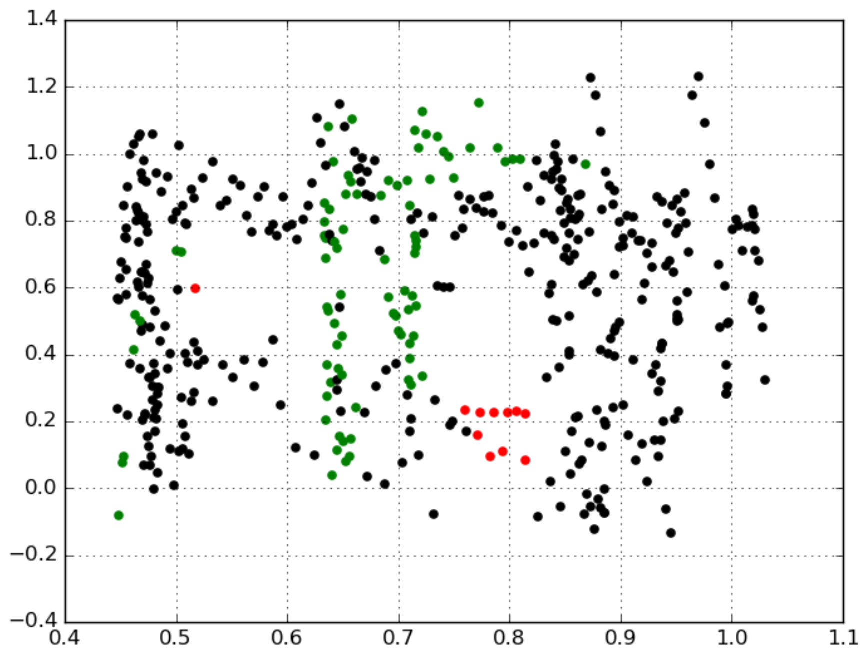

Fig. 5.3 Normalized SMA vs. BBP for the ML-based strategy after training. the X axis is the normalized SMA and Y axis is the BBP.

Part 6: Comparative Analysis

In part 6, I have updated all results based on out of sample period. First of all, orders book for rule-based method has to be re-generated again during out of sample period. Save it as `orders-rule-outsample.csv.`

Then, several steps are involved to build the ML trader:

Secondly, we use the same data in `data-training.csv,` generated from Part 4 to train the model.

Then, I use the RT Learner and Bag Learner to train to data in `data-training.csv` and get the model. The leaf size is set to be 5 and the number of bag is set to be 2. After training the data, we get predictions for Y during out of sample period. Then, we can get orders book named `orders-ML-outsample.csv` during out of sample period.

Next, the market simulator is used to get the table of values according to the orders book. Both orders book of Rule-based (`orders-rule-outsample.csv`) and ML-based method (`orders-ML-outsample.csv`) are fed into the market simulator to generate the following figure. Benchmark is the baseline.

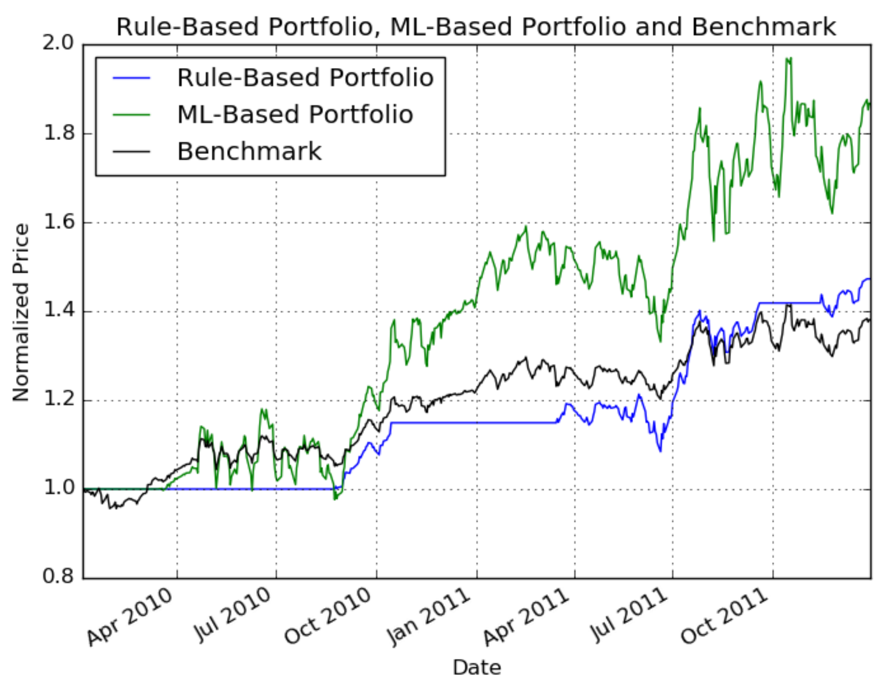

Fig. 6.1 Rule-based Porfolio, ML-based Potfilio v.s. benchmark price during out of sample period

From the Fig. 6.1, we can see that the ML-based portfolio shows the highest return while the benchmark has the lowest return. ML-based strategy outperforms the manual rule-based strategy. However, compared with the Fig. 4.1, we can see that performance during in sample period is slightly better than the performance during out of sample period. I summarize the performance of the stock, the manual strategy and the ML strategy or both in sample and out of sample periods in the following table.

From the table, we found that the performance of ML strategy during out of sample period is a little bit worse than that during in sample period. However, when we compare Fig. 4.1 with Fig. 6.1, we can see that the volatility for ML strategy during out of sample period is higher than that during in sample period. This is because we training the model using the in sample data. For Fig. 4.1, what we get is 'in sample evaluation' results while the results for Fig. 6.1 is 'out of sample evaluations.' Many causes would result in the performance degradation, such as over-fitting. Over-fitting issues would cause more 'errors' during out of sample period. To solve this problem, we could increase the number of bags. Note that here I only get small performance degradation thanks to multiple bags are used.

When we consider the 'best' factor, I think we have to consider volatility and return. So the parameters like Sharp Ratio would the best factor.

	Benchmark		Manual Strategy		ML Strategy	
Performance	In sample	Out of sample	In sample	Out of sample	In sample	Out of sample
Volatility	0.07	0.12	0.27	0.15	0.26	0.29
Mean	0.90	1.19	0.94	1.16	1.18	1.38
Cumulative Return	1.03	1.38	1.51	1.47	1.89	1.86
Sharp Ratio	288.6	222.6	78.2	173.6	101.9	106.8