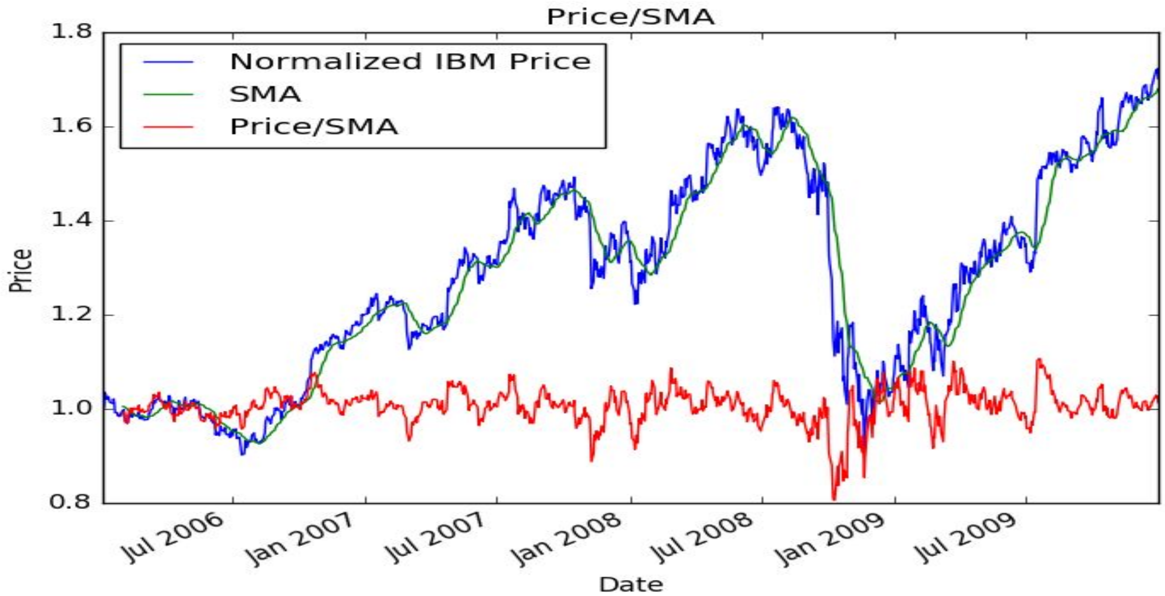
Part 1: Technical Indicators

1. Price/Simple moving average:

SMA is arithmetic moving average calculated by adding the closing prices of the stock for a number of time periods and then dividing this total by the number of time periods.

$$SMA_t = \frac{\sum_{i=t-window}^{t} closing\ price_i}{window}$$

where SMA_i is simple moving average on day i and $closing\ price_i$ is closing price of stock on day i. Indicator I used is $Price/SMA$ where Price is the price of the stock. When Price of Stock is going up then $Price/SMA$ indicator goes up and vice versa.



2. Bollinger Bands and BBP:

Bollinger bands consist of SMA of stock prices with window w, moving standard deviation of stock prices with window w, an upper band and a lower band.

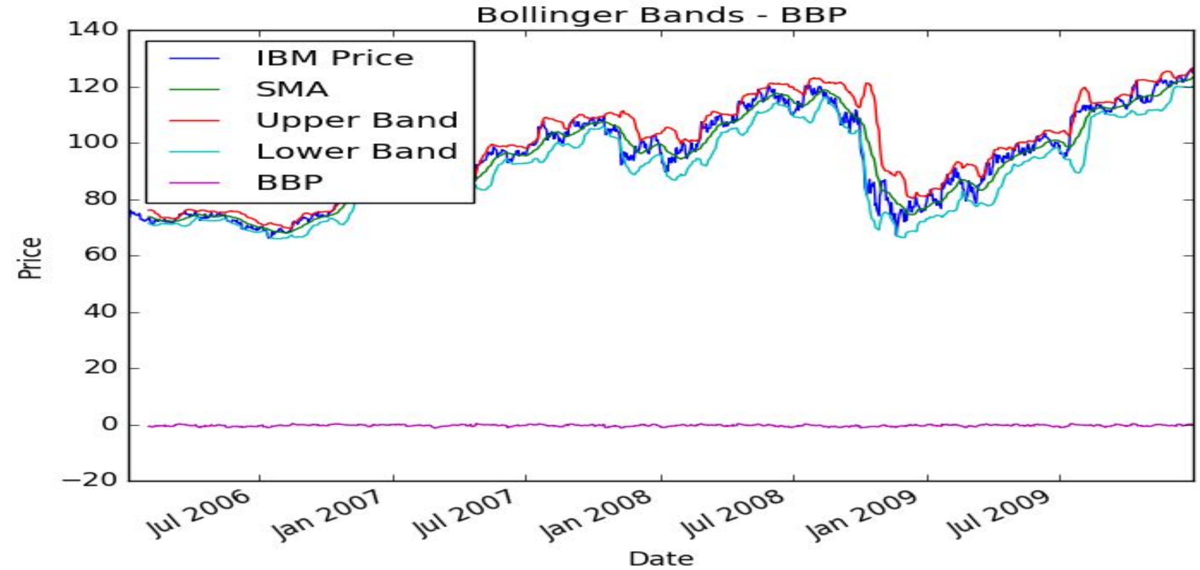
Upper Band = SMA + 2*Moving Std Dev; Lower Band = SMA - 2*Moving Std Dev

Moving Std Dev(i) = $\frac{\sum_{i=t-window}^{t} (price(i)-SMA(i))^2}{window}$ where moving std dev(i) is MSD for day i

Bollinger bands can be quantified using Bollinger Bands Percentile,

$$BBP = \frac{Price - lower\ band}{upper\ band - lower\ band}$$

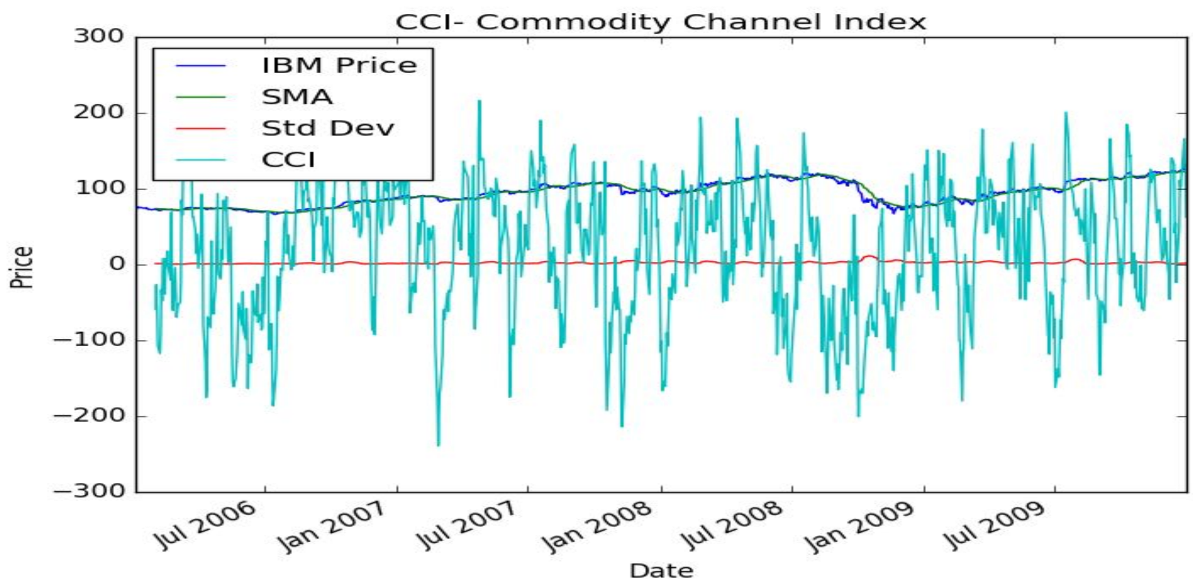
Bollinger Bands represent the volatility of stock. During rising volatility, the bands widen, and they contract as the volatility decreases. Prices are considered to be relatively high when they move above the upper band and relatively low when they go below the lower band.



3. CCI (Commodity Channel Index):

The CCI depicts relationship between price and a moving average.CCI can be calculated as

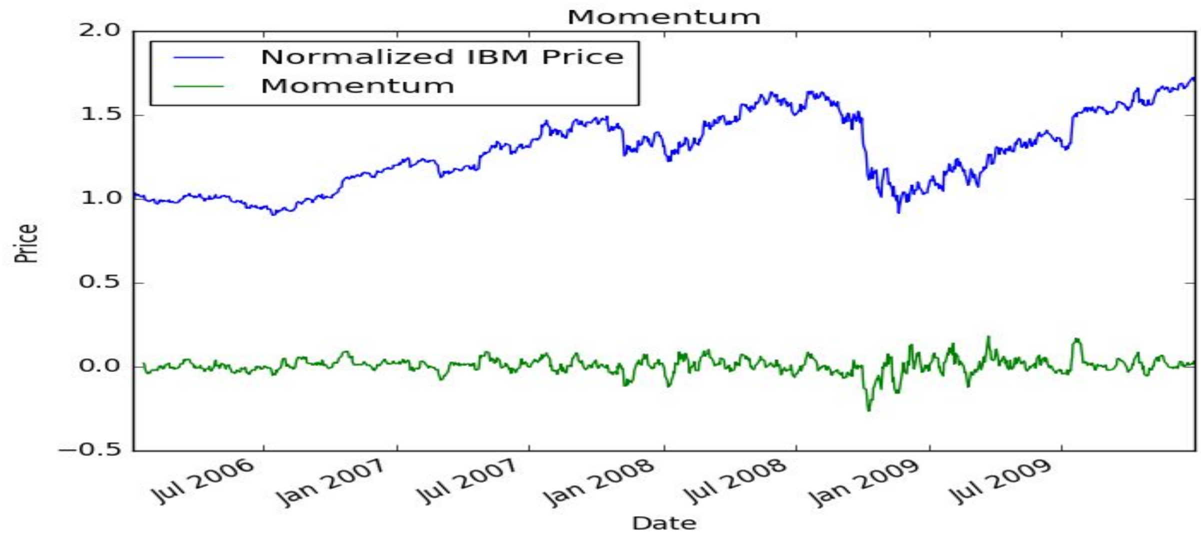
CCI(i)= $Closing\ Price\ of\ Stock(i)\ - SMA(i)/0.015 * Moving\ Std\ Deviation$



4. Momentum:

Momentum for window 'w' is given by: $Momentum(i)\ =\ \frac{Closing\ price(i)}{Closing\ Price(t-w)} - 1$

Momentum reflects price changes over a period of time. If momentum is greater than 1, then price will have increased over the window period and vice versa.



5. RSI (Relative Strength Index):

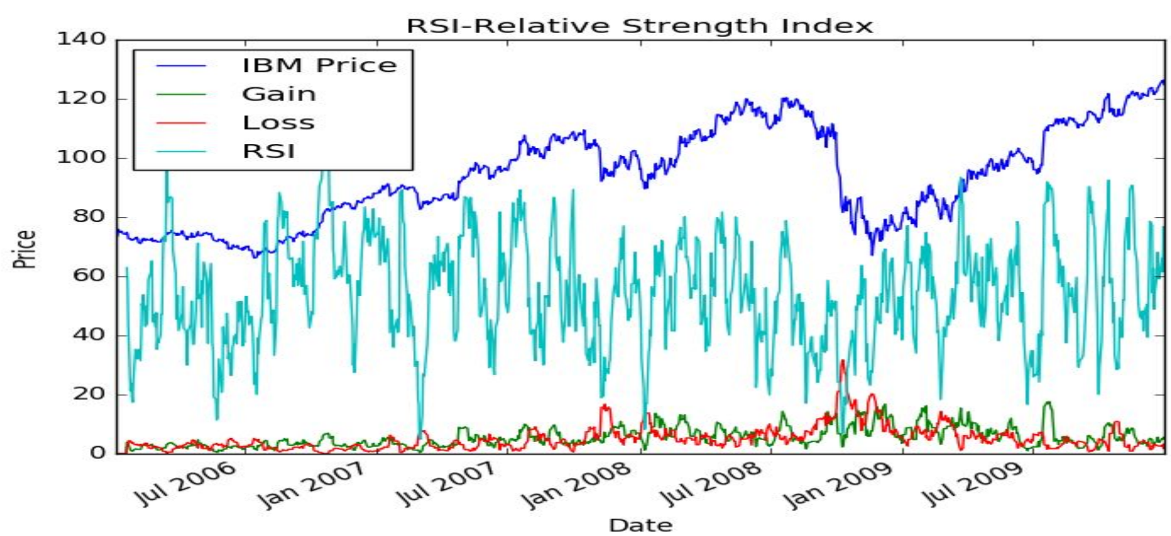
If RSI is greater than 70, it is considered overbought and when RSI less than 30, stock is considered oversold. RSI over window 'w' is calculated using formula:

$$RSI = 1 - \frac{100}{(1+RS)}$$

$$RS(i) = \frac{Avg\ gain(i)}{Avg\ loss(i)}$$

$$Avg\ gain(i) = \frac{\sum_{t=i-w}^{i} (price(t)/price(t-1)-1)\{if\ price(t)>price(t-1)\}}{w}$$

$$Avg\ loss(i) = \frac{\sum_{t=i-w}^{i} (price(t)/price(t-1)-1)\{if\ price(t)<price(t-1)\}}{w}$$



Part 2: Manual Rule Based Trader

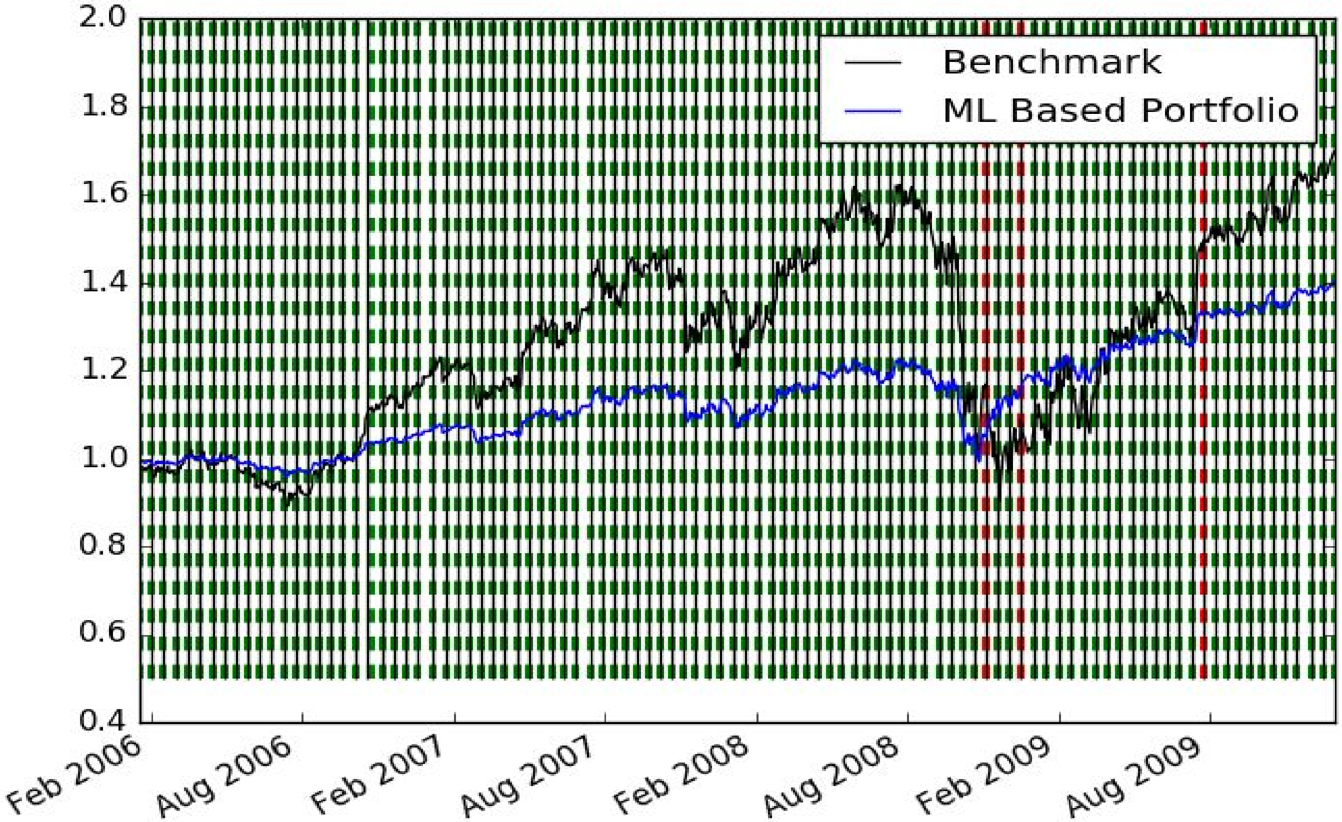
I am using the indicators mentioned in the above section to define the rule based order generation. I am using the following rules:

- If Price/SMA <1.04 then buy the stock and if Price/SMA >1.05 then sell the stock. If Price/SMA is high it means that today's price is higher than past prices, so the stock can be sold and vice versa
- If BBP(Bollinger Bands Percentile) is less than 0, then buy the stock and if BBP is greater than 1, then sell the stock. Bollinger Bands Percentile if less than 0, it means that the stock prices have reduced, so the stock can be bought and if BBP is greater than 0 implies the stock price increased so the stock can be sold.
- If CCI(Commodity Channel Index) is less than -50 then buy the stock and if CCI is greater than 150, then sell the stock.
- If (momentum_5 - momentum_10) is less than -0.1 then buy the stock and if (momentum_5 - momentum_10) is greater than 0.07 then sell the stock where momentum_5 is the momentum over window period 5 and momentum_10 is the momentum over window period 10. If momentum_5 is greater than momentum_10, it implies price has significantly increased over window period of 10, so the stock can be sold and if momentum_5 is lesser than momentum_10, it implies price has significantly decreased over window period of 10, so the stock can be bought.
- If RSI of IBM is less than 30 and RSI of SPY is greater than 30, then buy the stock. And if RSI of IBM is greater than 70 and RSI of SPY is less than 70, then sell the stock. IF RSI of SPY is less than 30, it means SPY is oversold and if RSI of IBM is less than 30, IBM is not oversold. So IBM stock can be bought and vice versa

I am using all the rules mentioned above and the ultimate final rule is:
If (Price/SMA <1.04) or (BBP <0) and (CCI <-50) and (RSI_IBM<30) and (RSI_SPY>30) and (momentum_5- momentum_10< -0.1), then buy the stock
Else if (Price/SMA >1.05) or (BBP >1) and (CCI >150) and (RSI_IBM>70) and (RSI_SPY<70) and (momentum_5- momentum_10>0.07), then sell the stock

Trading (buying/selling) stock is done with interval gaps of 10 days. If I traded today, I am not changing my position for next ten days and after 10 days, I am taking LONG (buy) or SHORT (sell) or do nothing position and I am holding maximum 500 stocks. And after every 10 day interval I am exiting my position.

The graph below shows the performance of my rule based order generation on training data set(i.e., on IBM stocks from 2006-01-01 to 2009-12-31). My rule based model is generating a cumulative return of 0.39.



Part 3: ML Rule Based Trader

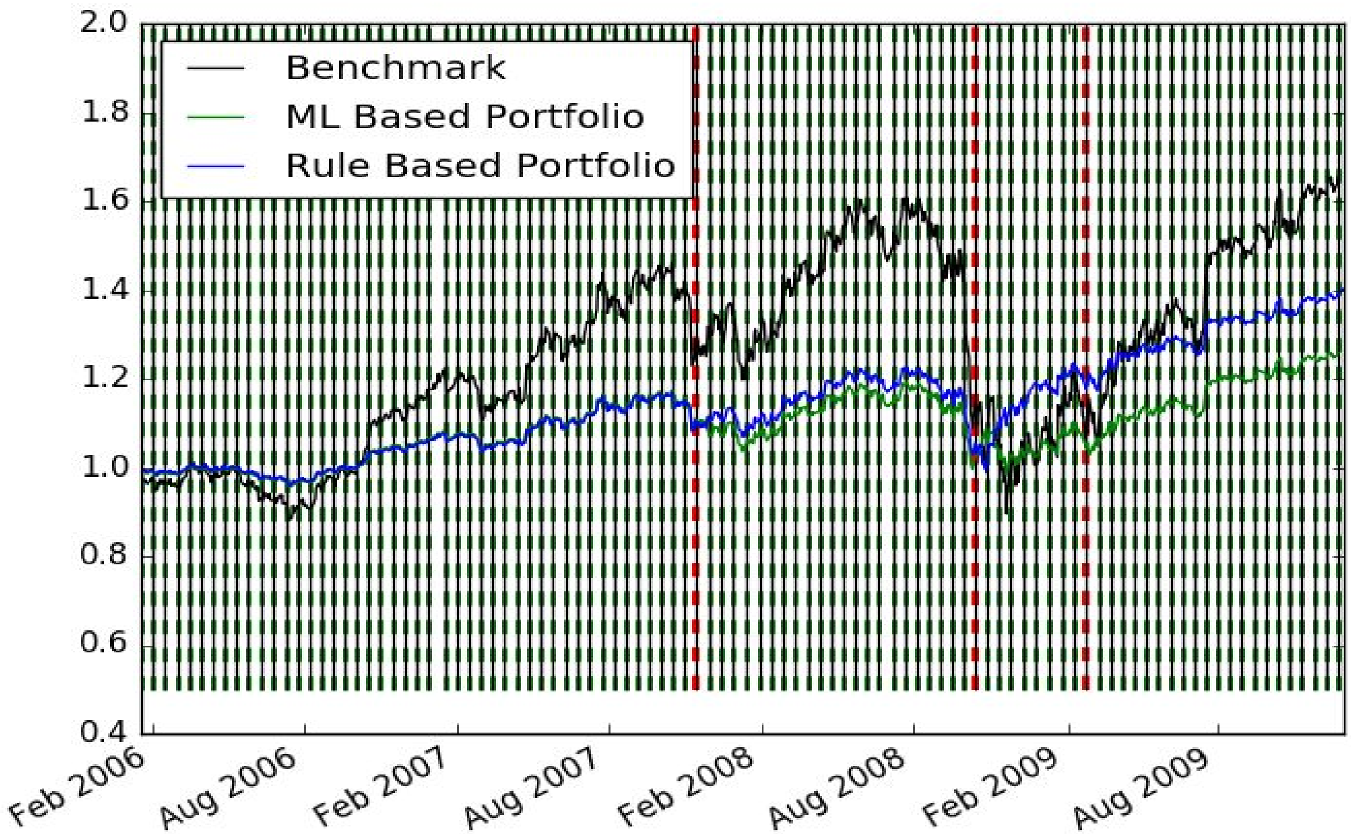
For ML Based Trader, I am using a features array as my training X data. I am calculating all the indicator mentioned in section 1 and placing them in an array and this array constitute my training parameters. For getting the training Y, I am using momentum calculated over window of 10 days. If momentum is less than -0.078, I can sell the stock i.e., I am setting Y for that day to be -1 and if momentum is greater than -0.088, I can buy the stock i.e., I am setting Y for that day to be 1 and 0 otherwise. In code form, it can be written as

```
ret = (price[t+10]/price[t]) - 1.0
if ret > -0.088:
    Y[t] = +1
else if ret <-0.078:
    Y[t] = -1
 else:
    Y[t] = 0
```

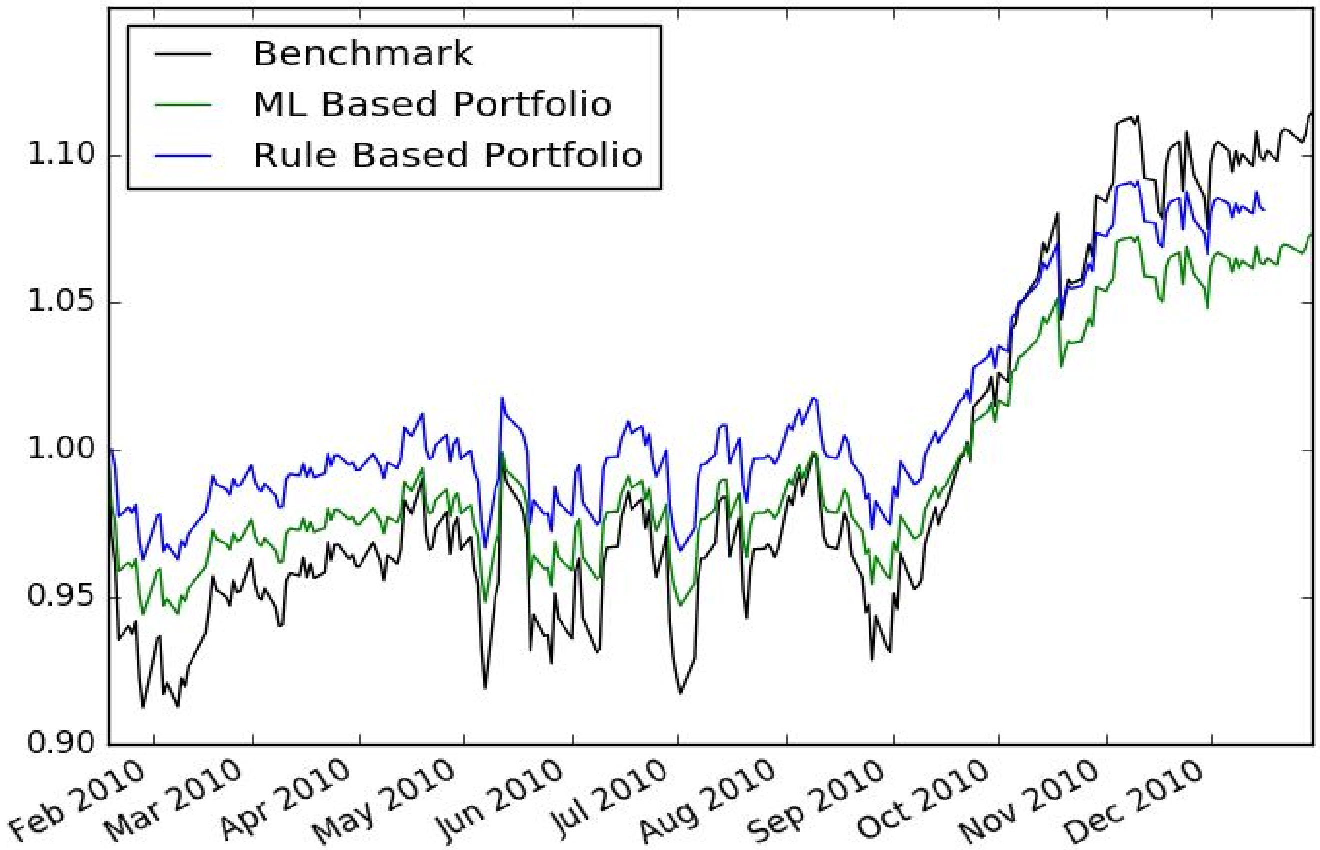
Now I have Xtrain(features array) and Ytrain(Y calculated using the above step). Now I will train my train my decision tree learner using the Xtrain and Ytrain. I am using decision tree with leaf size 5. For converting a Decision tree regression into classifier, once again I am using the rule mentioned above i.e, I am taking the mean of Y of all leaf nodes and if the mean is greater than -0.088, then I am classifying the input into +1 class and if mean is less than -0.078, input is classified into -1 class and 0 otherwise. Once again, I am trading (buying/selling) stock with interval of 10 days. If I traded today, I am not changing my position for next ten days and after 10 days, I am taking LONG (buy) or SHORT (sell) or do nothing position and I am holding maximum 500 stocks. And after every 10 day interval I am exiting my position.

The graph below shows the performance of my ML based order generation on training data set(i.e., on IBM stocks from 2006-01-01 to 2009-12-31). My rule based model is generating a cumulative return of 0.26.

My Rule based learner is performing better than the ML based model because, in designing ML based model, Y train (the matrix that decides the orders) is built using 10 day returns. But in rule based model, the orders are decided based on four other factors (section 1) along with momentum_10 which make rule based system better.



Part 4: Comparative Analysis



Comparison table:

	Stock Performance	ML Based Model	Rule Based Model
In sample cumulative return	0.7	0.23-0.28	0.39
Out of sample cumulative return	0.13	0.073	0.085

Performance: out of sample Vs in sample:

For both ML based model and Rule based model, the out of sample performance is relatively lesser than the in sample performance. For ML Based model, changes in the price in out of Sample data would not follow the exact same pattern as in sample data. When data is being trained, the BUY and SELL triggers are computed based on behavior of In sample data so clearly model would perform better on in sample data than out of sample data. Similarly for rule based model, the rules to generate order files are devised in a way that best fits the price variation pattern of in sample data, so when used the same rules for out of sample data, the performance will be not as good as in sample's because the probability that price changes in out of sample data follow exact same rules will be less.

Performance of Manual strategies Vs ML strategies:

My manual strategies are resulting in better performance compared to ML strategies for both in sample data as well as out of sample data. This is because, in designing the ML based model which is build using Xtrain and Ytrain, Ytrain is built using 10 day returns (momentum_10) feature. Whereas, in the rule based system, the rules are built using four other indicators(mentioned in section 1) which makes the rule based system perform better than the ML based model.

Manual strategy is not susceptible to the flaw of designing model using less number of features in deciding the Y because manual strategy is using more number of features in devising the rules to generate orders.