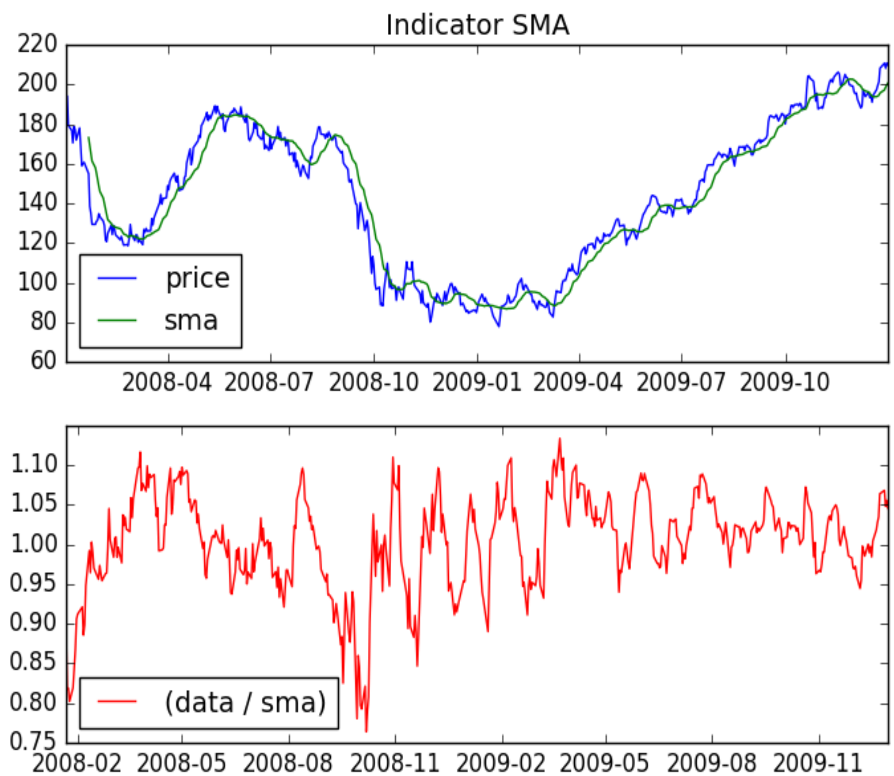
Part 1: Technical Indicators:

Simple moving average: Simple moving average is moving average calculated by dividing the sum of a set of closing prices by the total number of prices found in the series. It is the straight forward calculations that calculate the average price chosen over a time period. This is used b y traders/analysts operating on longer time frames, such as daily or weekly charts versus hourly charts. In the following formula,

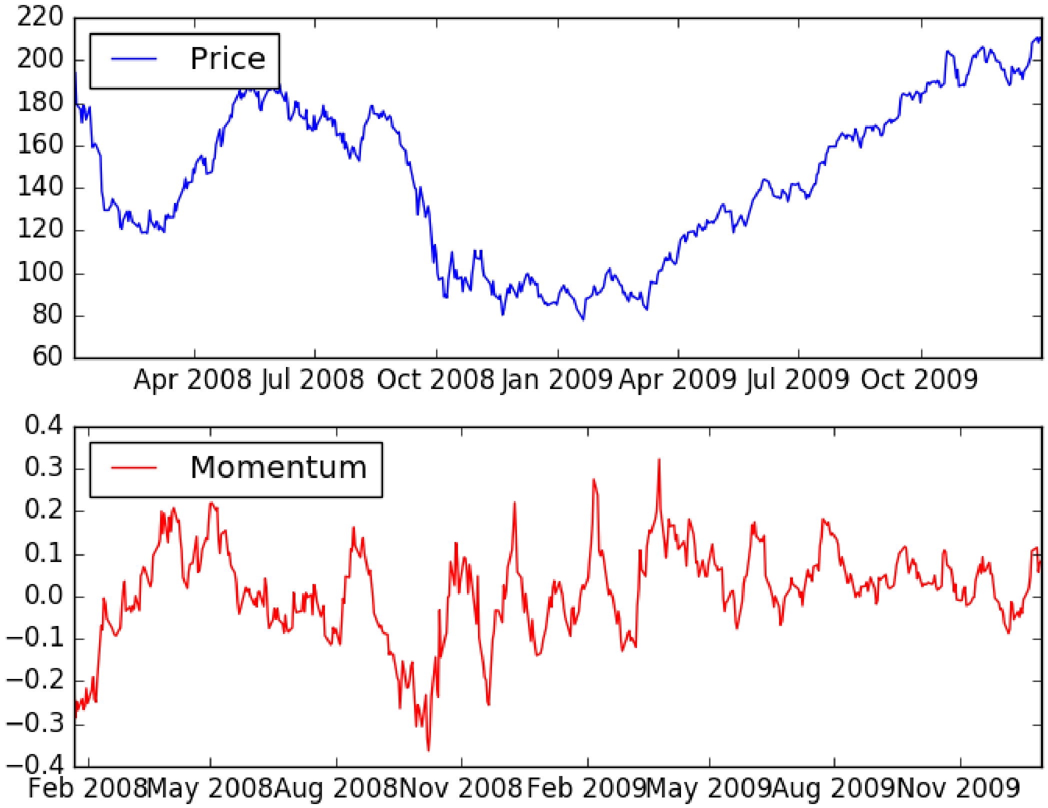
$\mathbf{SMA_t = \sum_{i=t\text{-}window}^{t}}$ ***Closing price/window*** where SMA_t is simple moving average on day i and closing $price_i$ is closing price of stock on day i.



1. **<u>Momentum</u>**:

   Momentum is the measurement of the speed of price changes over a period of time in which it continuously takes the price differences for a fixed time interval. It measures the rate of rise or fall in stock prices; hence helps recognize the trendlines.

   Momentum (i)= [Closing price (i)/Closing price (t-w)]-1. If momentum is greater than 1, then price will have increased over the window period (w)and vise versa.

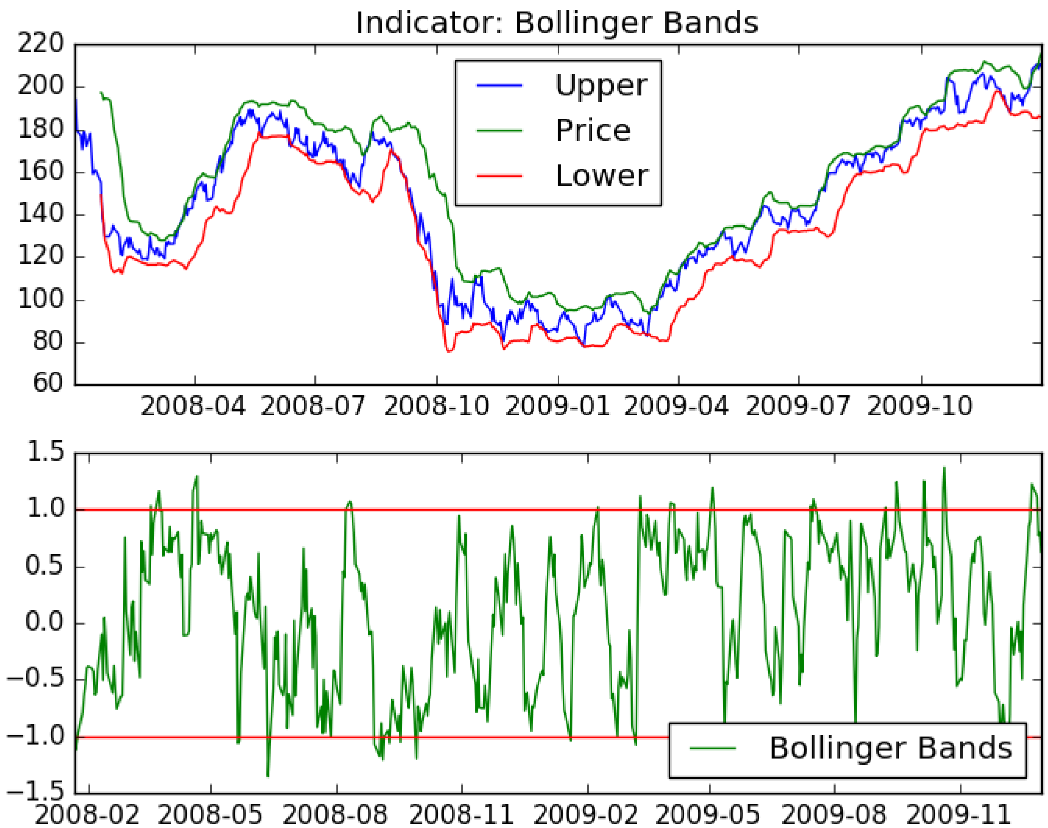


2. **Bollinger Band**:

Bollinger Band that is quantified as Bollinger Bands Percentile is calculated by dividing the difference between Price and lower band by the difference between upper band to lower band, where lower band is equals to SMA – 2*Moving Std Dev and Upper band is equals to SMA+2*Moving Std Dev. Standard deviation represents the volatility of stock where the markets become more volatile, the bands widen; during less volatile periods, the bands contract. Any breakout above or below the bands is a major event.

Formula:

BBP= (Price-Lower Band)/ (Upper Band- Lower Band)

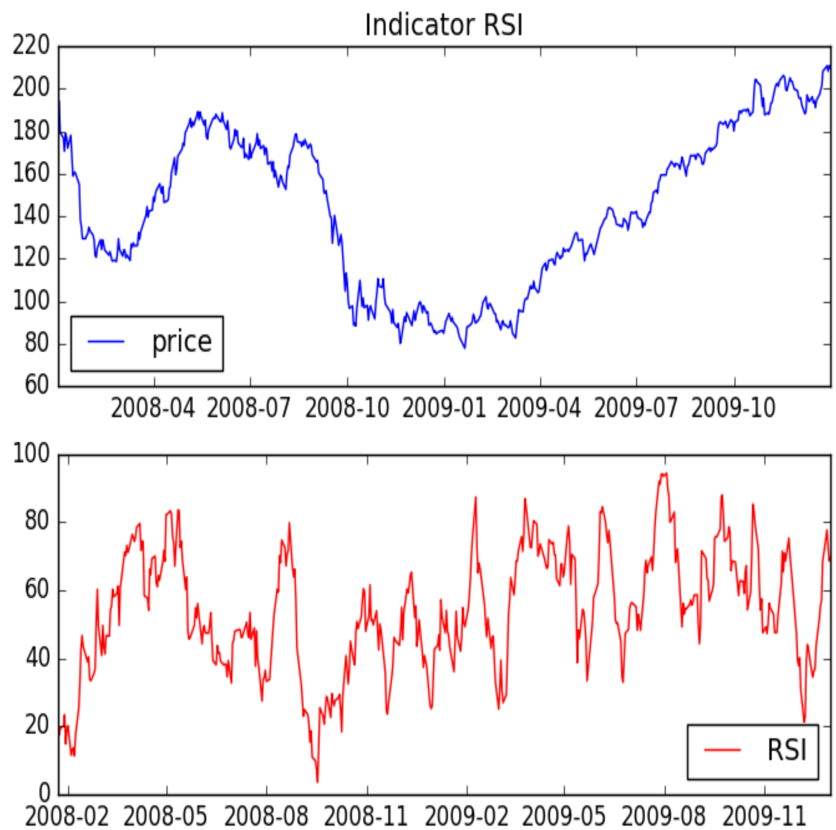


3. <u>RSI</u>

Relative Strength index is a momentum indicator that compares the strength of gains and losses over time period to measure speed of change in price movements. It attempts to identify overbought (RSI >70) or oversold (RSI <30) conditions in the trading of an asset.

RSI= (100-100/(1+RS))/100

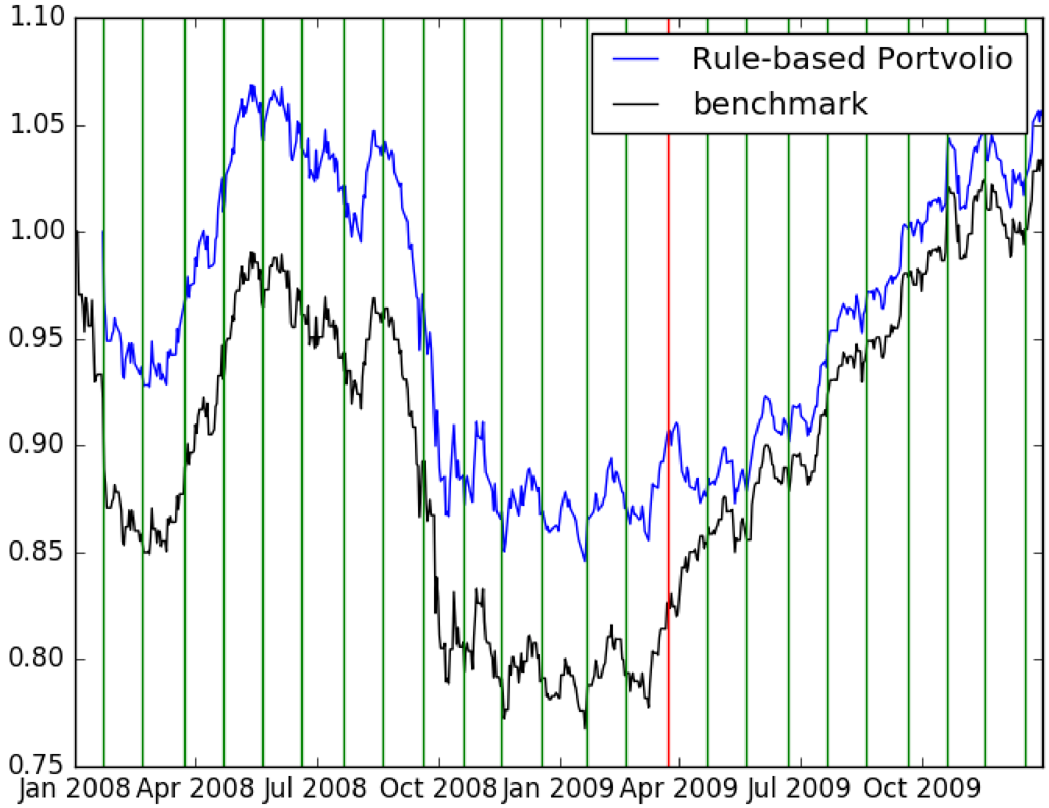
RS(i)= (Avg gain (i))/(Avg loss (i)



Part 2: Best Possible Strategy

Skipped

Part 3: Manual Rule-Based Trader



The overall strategy of the manual trading rules is buy when the stock appears to be oversold relative to the market and to sell when the stock appears to be overbought relative to the market. A low price/SMA ratio, low Bollinger Band percentage, and low RSI value all indicate that the stock is oversold. In addition, a low momentum value indicates that the stock is also trending downward at the time, making it even more likely that the stock is oversold at that time and will continue in that direction.

I'm using the indicators mentioned in the above section to define the rule. Here's the overview of the indicator's rule:

If [(bbp < 0.49) and (rsi < 30) and (mom< -0.059) or (Price/SMA <1.1)] →buy/Long Apple stock

ElseIf [(bbp > 0.5) and (rsi > 70) and (mom> 0.051) or (Price/SMA > 1)] →Sell/short Apple stock

The following are the explanations of each indicator rules:

- If [Price/SMA < 1.1] then buy the stock and if [Price/SMA >1.05] then sell the stock. When the Price/SMA value is higher, which means that price is increasing compare to past prices, in this situation, we can sell the stock or vice versa.

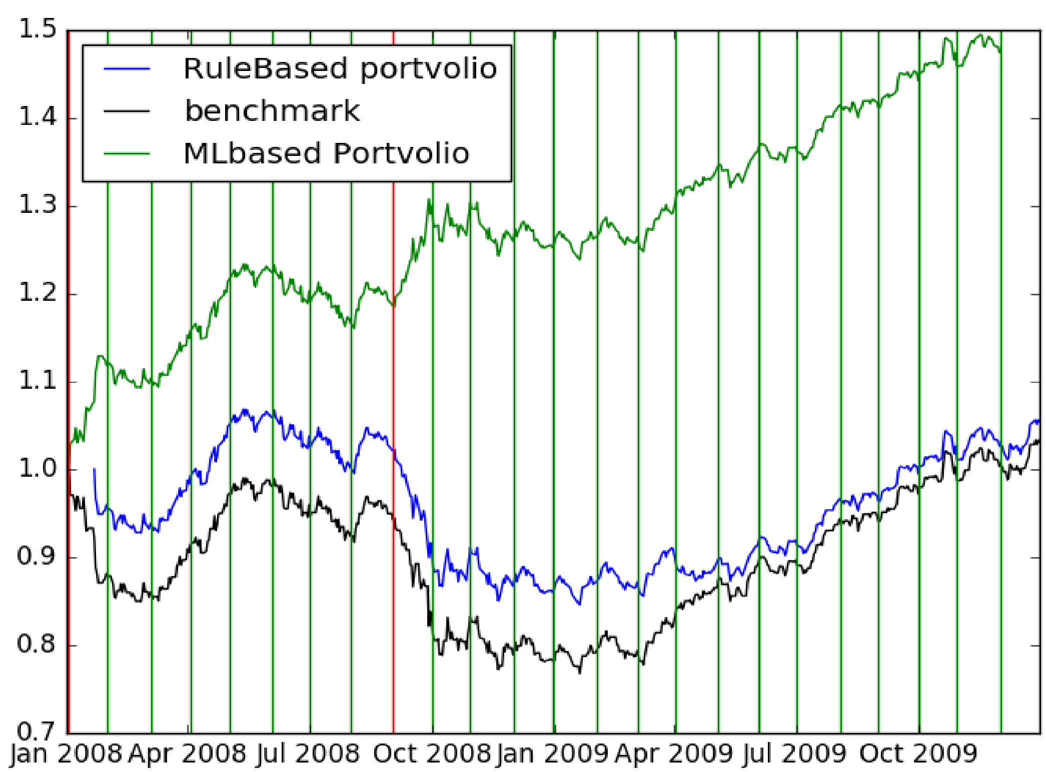
- If Bollinger Bands Percentile(BBP) is less than 0.49, then buy the stock whereas if BBP is greater than 0.5, then sell the stock. If a BBP value is 0.49, which means the stock prices have reduced, the stock can be bought. However, stocks can be sold when the BBP value is greater than 0.5

- If the RSI is less than 30, then buy the stock. The stock can be sold when the RSI is greater than 70. if RSI of the Apple is less than 30, apple stocks are oversold. It is better time to buy those stocks.

- If the momentum value is less than *-0.059*, the stock is likely oversold and there is a likely a divergence from the market so go LONG. On the other hand, if the momentum value is greater than 0.051, the stock is overbought and there's a likely a divergence from the market so go SHORT

I've used nested if else statements to identify whether a Long/Buy or Short/Sell condition is met. Trade are performed with interval of 21 days as per project requirement. In other words, if today is first trading day, the position must not be changed for until next 21 days. At the end, the order list is generated and saved as orders_RuleBased.csv file. Marketsim.py takes the order file as an argument and compute portfolio values.

Part 4: ML Trader



For ML, Based Trader, *trainX* values are calculated each day from the current day's (and earlier) data, but the *trainY* value (classification) is calculated using data from the future.

- I've normalize the indicators so that so that they have zero mean and standard deviation.
- I've calculated all the indicator mentioned in part 1 and merged them into a dataframe. The dataframe is my training *trainX* data.
- Get the *trainY* training data from the 21 day returns for each day over the training dates. If the return is greater than -0.078 then set *trainY* to buy or 1 and if the return is less than -0.078 then set the *trainY* to sell or -1

```
Y[t] = 0 ret = (price[t+21]/price[t]) - 1.0
if ret > YBUY:
   Y[t] = +1 # LONG
else if ret < YSELL:
   Y[t] = -1 # SHORT
else:
   Y[t] = 0 #Do Nothing
```

- Now, I have trainX and trainY. Train and test your learning strategy over the in-sample period using BagLearner with leaf size is 5 and 15 bags. It should be noted that the values of leaf size and bagging were tweaked, in my case, BagLearner performed better than the RTLearner.
- Get predicted y values by querying trainer with x (*testX*) testing data
- Convert predicted y values to buy or sell values by rounding up from .5 to 1 and down from -.5 to -1. Everything else is set to 0.
- Ensure that orders are held for at least 21 days before exiting position
- Then diff the orders so that there is only an order when target shares change.

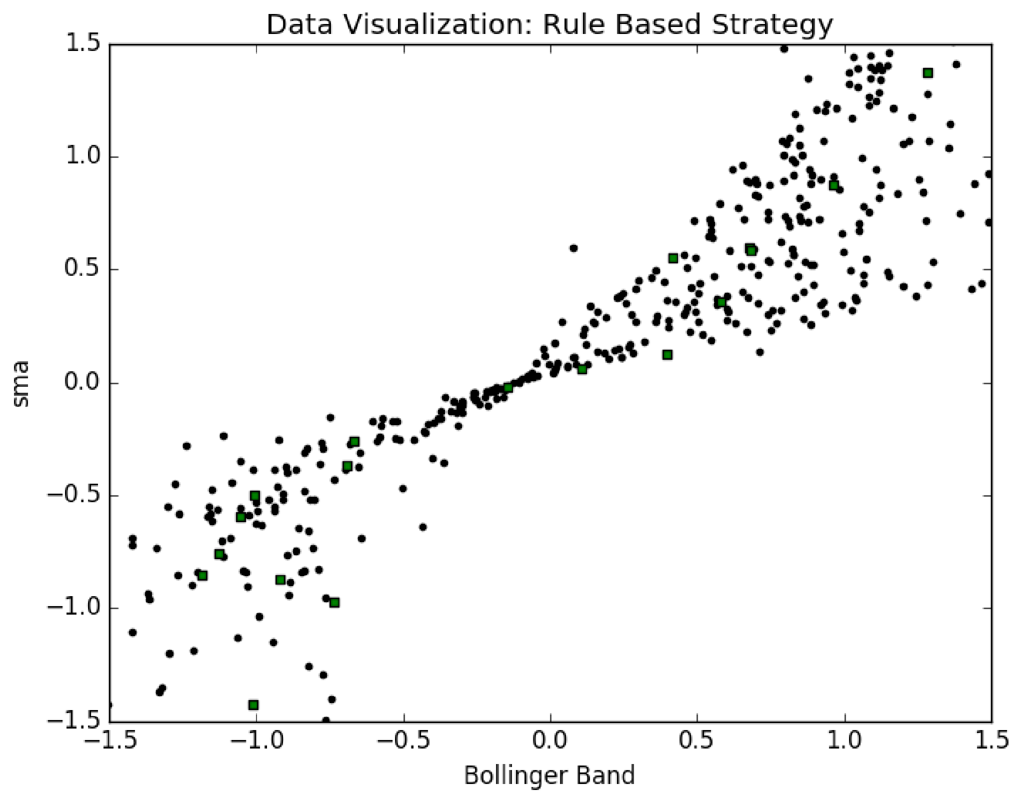
At the end, the order list is generated and saved as ordersMLBased.csv file. Marketsim.py takes the order file as an argument and compute portfolio values. The following table are generate using within sample/training period data. It should be noted that ML based cumulative return is highest followed by rule based and benchmark.

Type	**ML Based**	**benchmark**	**Rule Based**
Sharpe Ratio	2.05790798	0.15105699	0.27951807
Volatility (stdev of daily returns)	0.00636364	0.00726051	0.00793402
Average Daily Return	0.00082496	6.90888150e-05	0.0001397
Cumulative Return	0.47488	0.03164	0.05456

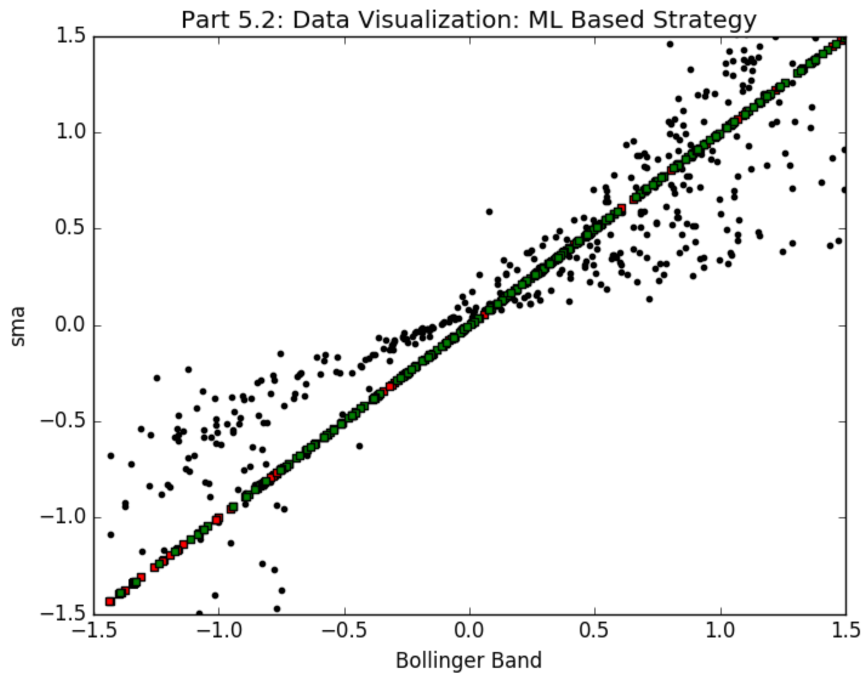
Part 5: Visualization of data

Bollinger band and Simple moving average are two indicators that I used to plot the following charts.

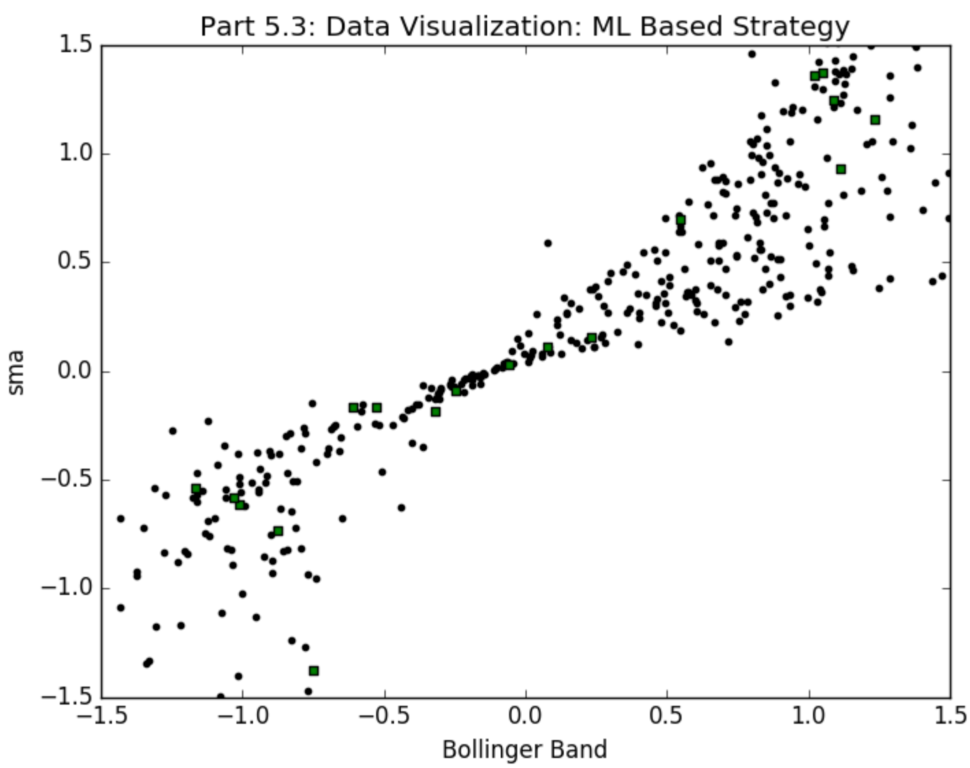
1. Your rule-based strategy.



2. Training data for your ML strategy



3. Response if your learner when queried with the training data



Part 6: Comparative Analysis (10%)

The following graph is generated using out of sample data. It is noted that I haven't train or tweak my learner on this data. However, I've used the classification learned using the training data.

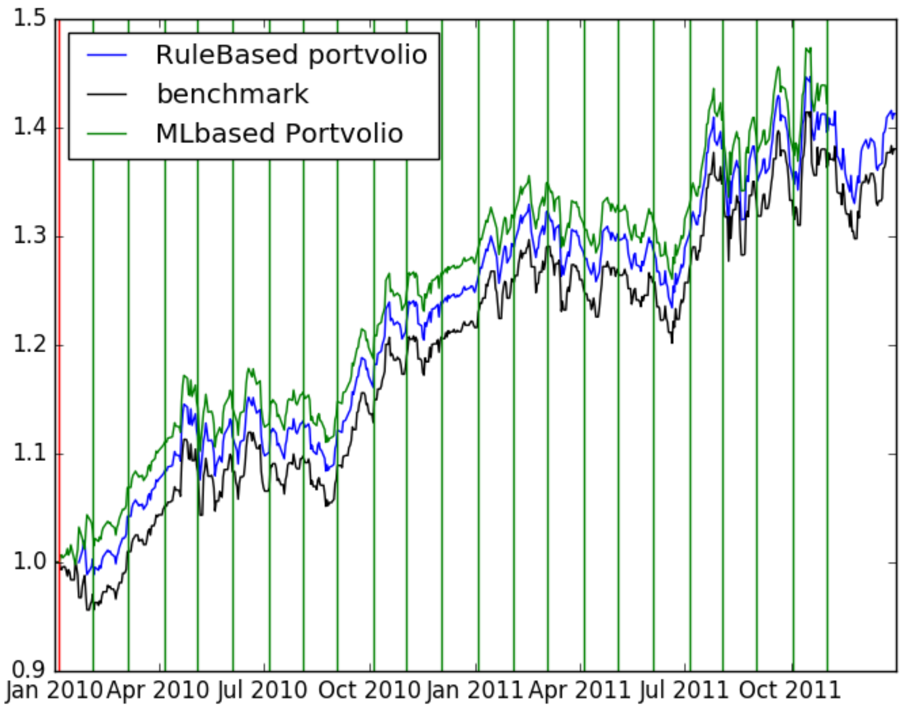


Table 6.1: Out of sample summary table

Statistics	ML Based	benchmark	Manual Based
Sharpe Ratio	1.55016383	1.04376183	1.41303443
Volatility (stdev of daily returns)	0.00815057	0.00710974	0.00831193
Average Daily Return	0.00079591	0.00046747	0.00073987
Cumulative Return	0.4223	0.38034	0.41274

Table 6.2: In sample summary table

Statistics	ML Based	benchmark	Rule Based
Sharpe Ratio	2.05790798	0.15105699	0.27951807
Volatility (stdev of daily returns)	0.00636364	0.00726051	0.00793402
Average Daily Return	0.00082496	6.90888150e-05	0.0001397
Cumulative Return	0.47488	0.03164	0.05456

Performance: out of sample Vs in sample:

My out of sample performance on ML based model is relatively lesser than in sample performance. However, Manual based model performed better than in in-sample data than in out of sample data.

For ML, based model, changes in the price in out-of-sample data would not follow the exact same pattern as in training 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. Hence, it occurs overfitting. And reason of overfitting can be that the model was tweaked to get maximum performance on the training data set.

For rule based model, Manual based model performed better than in in-sample data than in out of sample data. This model is built by using indicators and its rules. These rules are made by combination of indicators calculations. These indicators are proven method of examining past market data to help forecast future price movements. I think, the reason this model performed better is because indicators rules are less likely changes regardless or in or out sample data.

Performance: Comparison within Out of sample:

In out-of-sample data, ML based model is relatively better than others.