David Vinegar
dvinegar3
****Charts begin on page 3**

Describe each of the indicators you have selected in enough detail that someone else could reproduce them in code.

Indicator 1: 3 day Momentum
momentum[t] = (price[t]/price[t-N]) - 1
The first indicator used by my learner is three day momentum. It is calculated simply by dividing the price of the stock by its price three days ago and subtracting one.

Indicator 2: Volatility
The second indicator, volatility, is calculated by taking the five day rolling standard deviation of the stock's daily returns. Daily returns are calculated by taking what the stock is worth today, subtracting what it was worth yesterday, and dividing by what it was worth yesterday. The standard deviation of the daily returns was calculated by Pandas using its rolling_std() function. The rolling_std() function looks at the past n observations (where n=5 days in my scenario), calculates a mean of those observations, sums the squared difference between each observation and the mean, and then takes the square root of that sum.

Indicator 3: Bollinger Bands
bb_value = (stockPrices - SimpleMovingAverage) / (2 * StandardDevOfReturns)
The third indicator, bollinger bands, is calculated by finding the difference between the price of the stock and its simple moving average, and then dividing that difference by two times the standard deviation of the stocks returns. Standard deviation of returns is calculated exactly as it was for the volatility indicator. The simple moving average is calculated by determining the average stock price in the past three days.

Describe your trading policy clearly.

The learner iterates through each day of the trading period. On each day, it determines if the predicted price is greater than the current price. If it is, a long position is opened and any open short position is closed. If the predicted price is less than the current price, then a short position is opened and any long position is closed. The position is held for five days no matter what. After the five day period, the learner repeats itself by asking which direction the price is going in, and opening up a long or short position as described above.

Discussion of results. Did it work well? Why? What would you do differently?

Overall the strategy worked well, but not great. The results for ML4T-220 were up over 700% both in and out of sample. Results for IBM in sample were up 80% in sample vs about 22% if you had just bought IBM outright and held it for the same time period. Out of sample for IBM, the algorithm was up about 47%, which is just slightly better than the 46% you could have gained if you had bought and held IBM for the same out of sample time period.

The strategy also performed well as measured by correlation and RMSE, except out of sample. Here are the statistics:

ML4T-220 Statistics
In Sample RMSE: 0.000777897743405

In Sample Correlation: 0.99914374593

Out of Sample RMSE: 0.000329759194169
Out of Sample Correlation: 0.999845238546

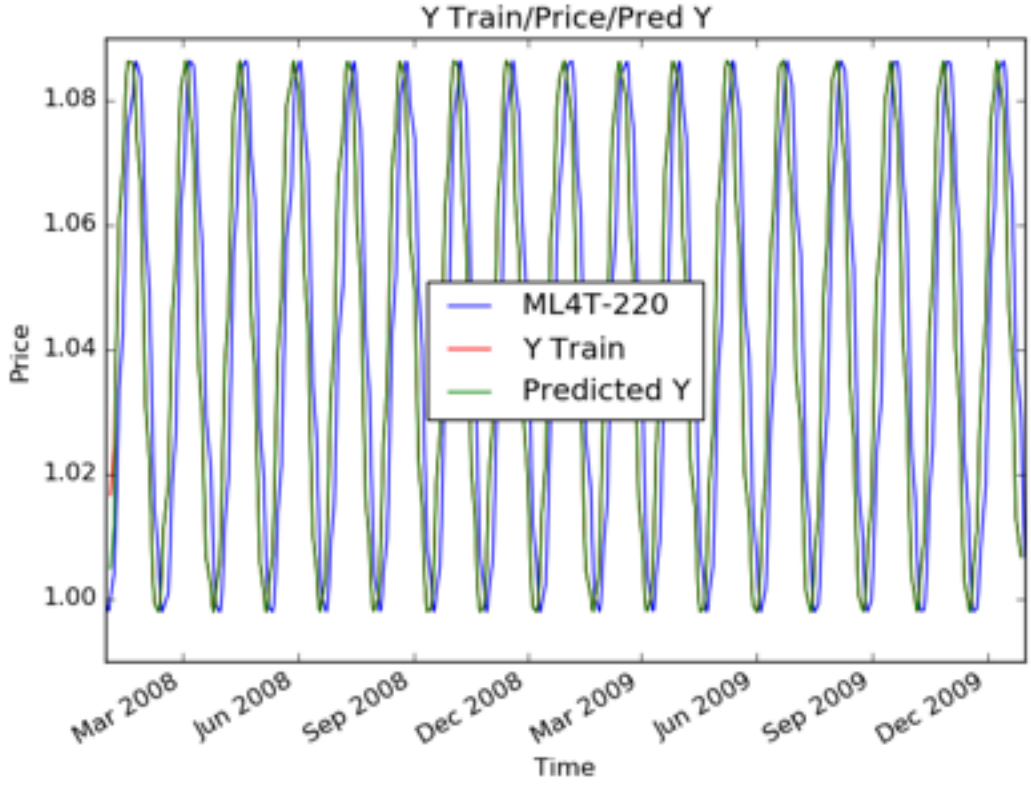
IBM Statistics
In Sample RMSE: 0.0237231559148
In Sample Correlation: 0.70170007879

Out of Sample RMSE: 0.0276540363054
Out of Sample Correlation: 0.034098094607

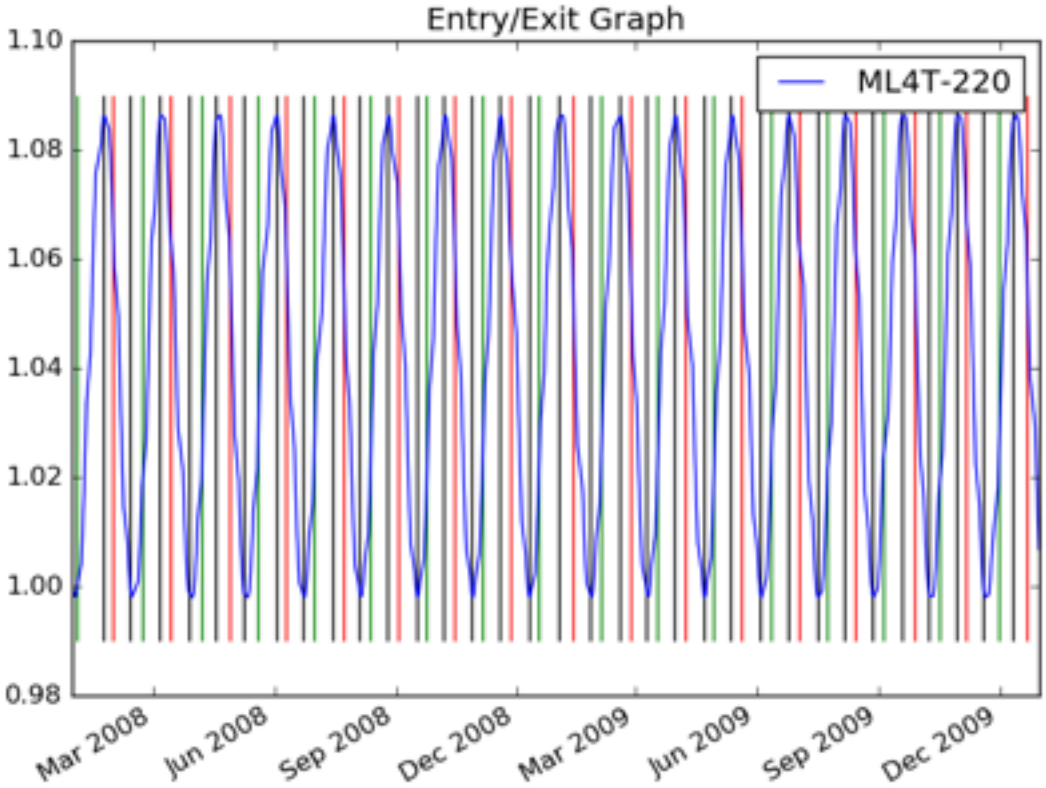
Why did the learner work well? Firstly, it did not work well out of sample. It barely beat the benchmark buy and hold IBM strategy. Why did it perform well on the other tests? This question is best addressed one test at a time. The strategy performed excellent on ML4T-220 because it is a constantly repeating pattern. This kind of pattern is easy for a KNN learner, which I used.
The strategy performed well on IBM in sample more or less because it was in sample. There are three unique data points that don't happen frequently. In other words, what are the odds that within the same in sample period, the same volatility, momentum, and bollinger value will all crop up? Because those chances are so slim, and my KNN k value is equal to two, it should pretty much always find out the exact correct point on the training values to go to, along with one other value with similar x variables.

It is debatable whether or not the strategy worked on well out of sample for IBM. On the one hand, it went up 47% in a 2 year time frame, a solid return. On the other hand, as stated previously, it nearly didn't do as well as a buy and hold strategy, which would encounter much lower commission fees. I think the strength of its performance speaks to the utility of the indicators I chose. Volatility is very useful. If the stock is very volatile over a five day span, I think you can make a smart guess that the next five days may very well be volatile. Stocks are usually more volatile in the downward direction. Momentum is also very useful. If a stock is having a decent three days, it may more often than not be a sign of a positive long term trend.

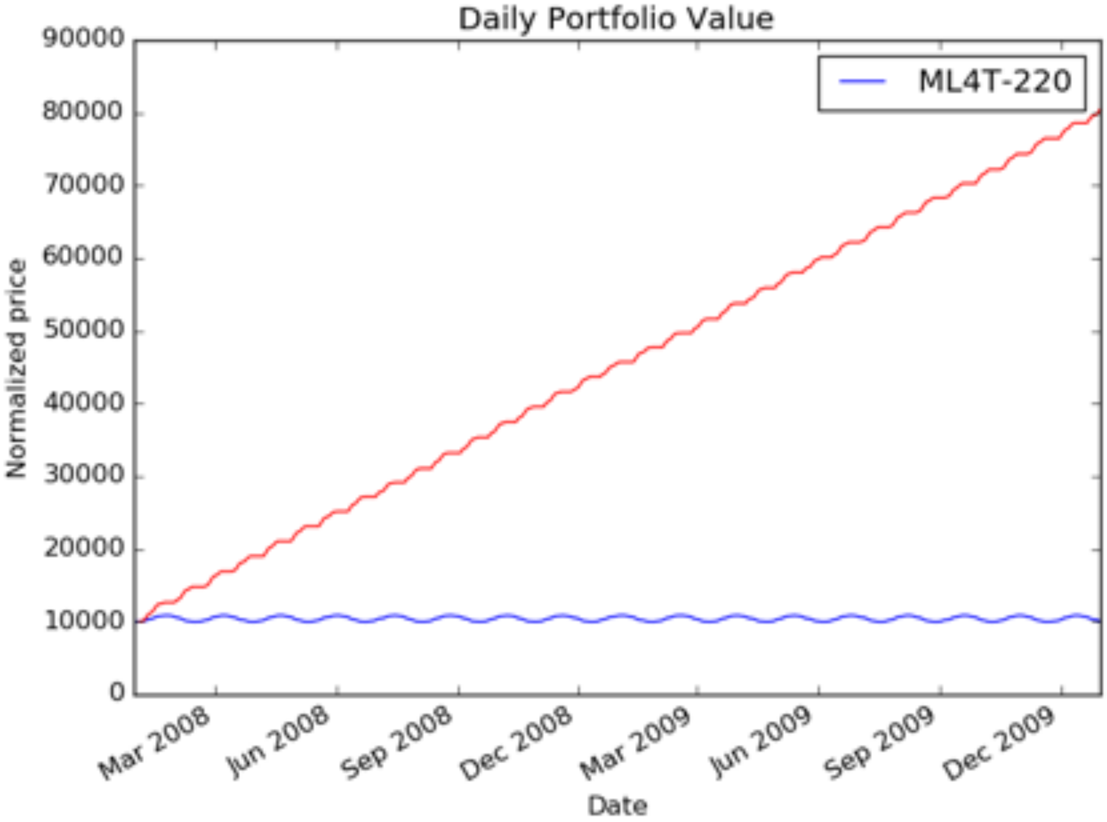
Given more time and resources, I would add more indicators. I would like to include S&P momentum. Many stocks move in high correlation with the S&P 500.



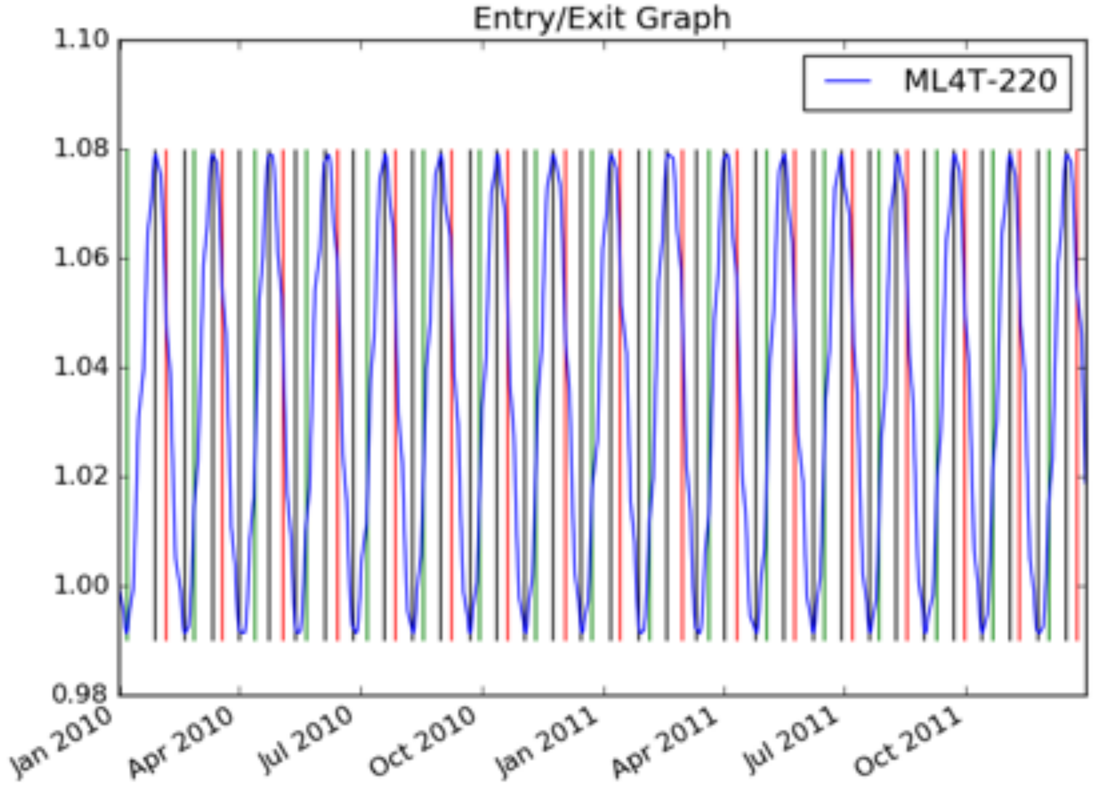
In the graph above (In sample Y train/Price/ Pred Y), the predicted values lie mostly on top of the training values.



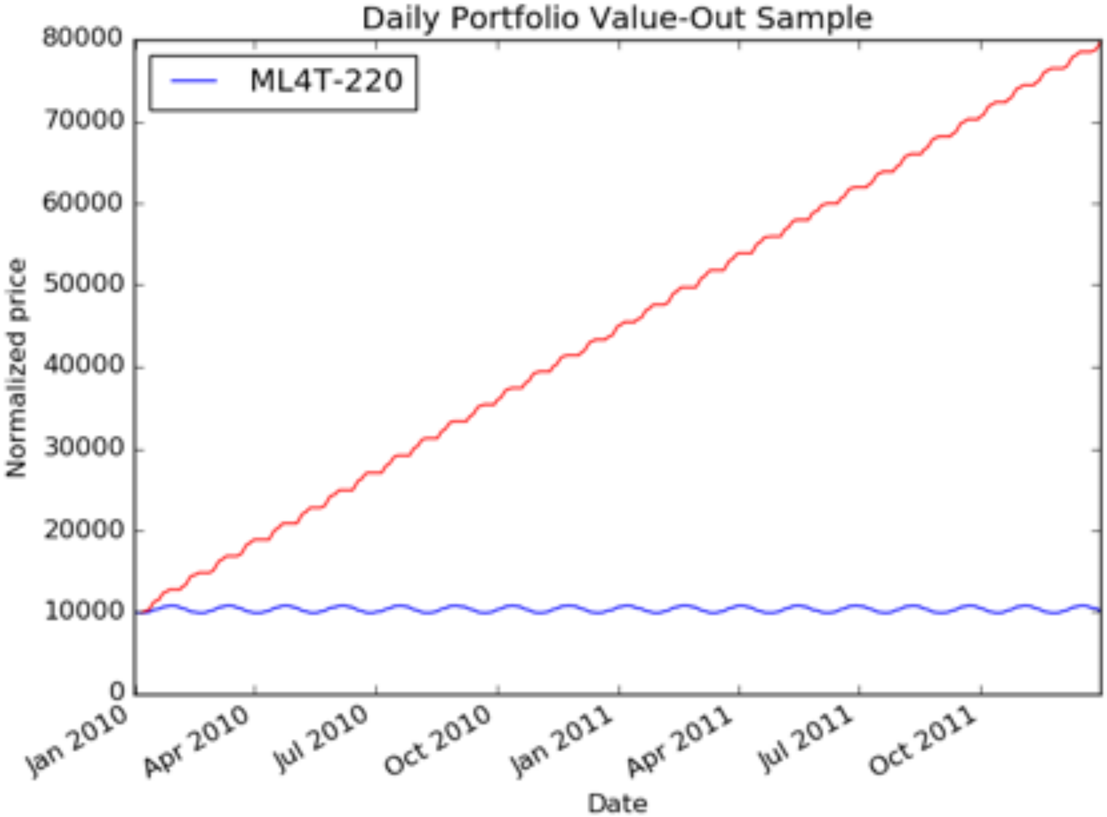
Above is the in sample entry and exit point graph of ML4T-220.



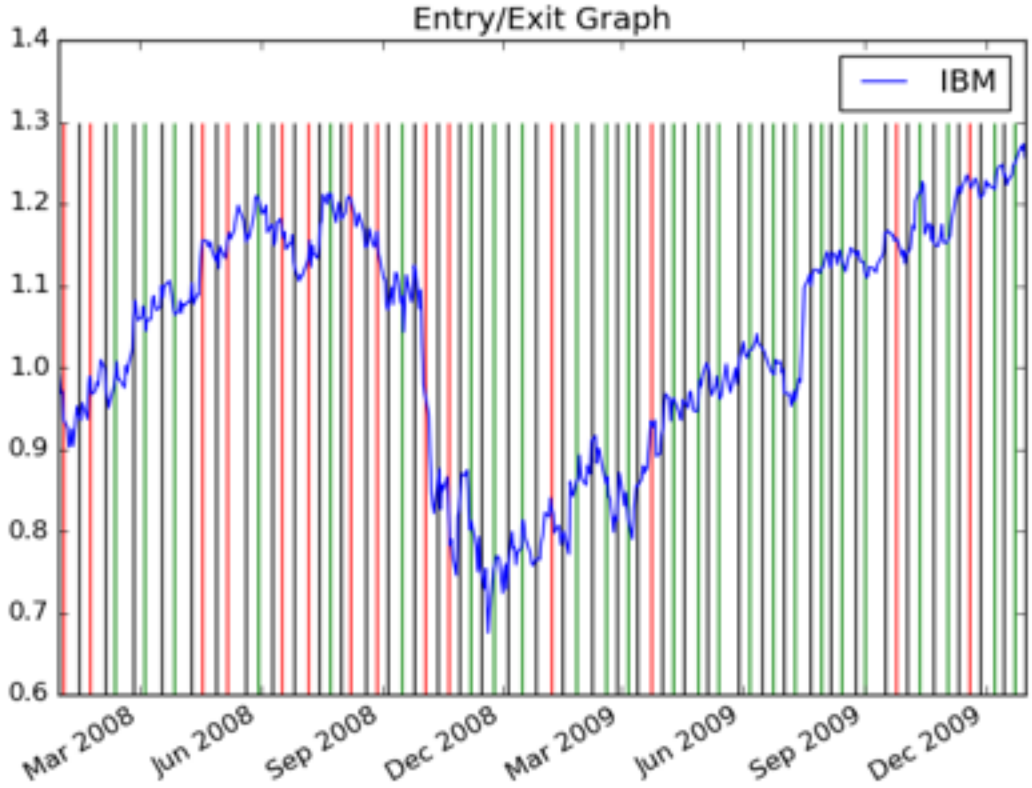
Above is the in sample portfolio backtest of the learner on ML4T-220. The learner's performance is in red, while ML4T-220 is in blue.



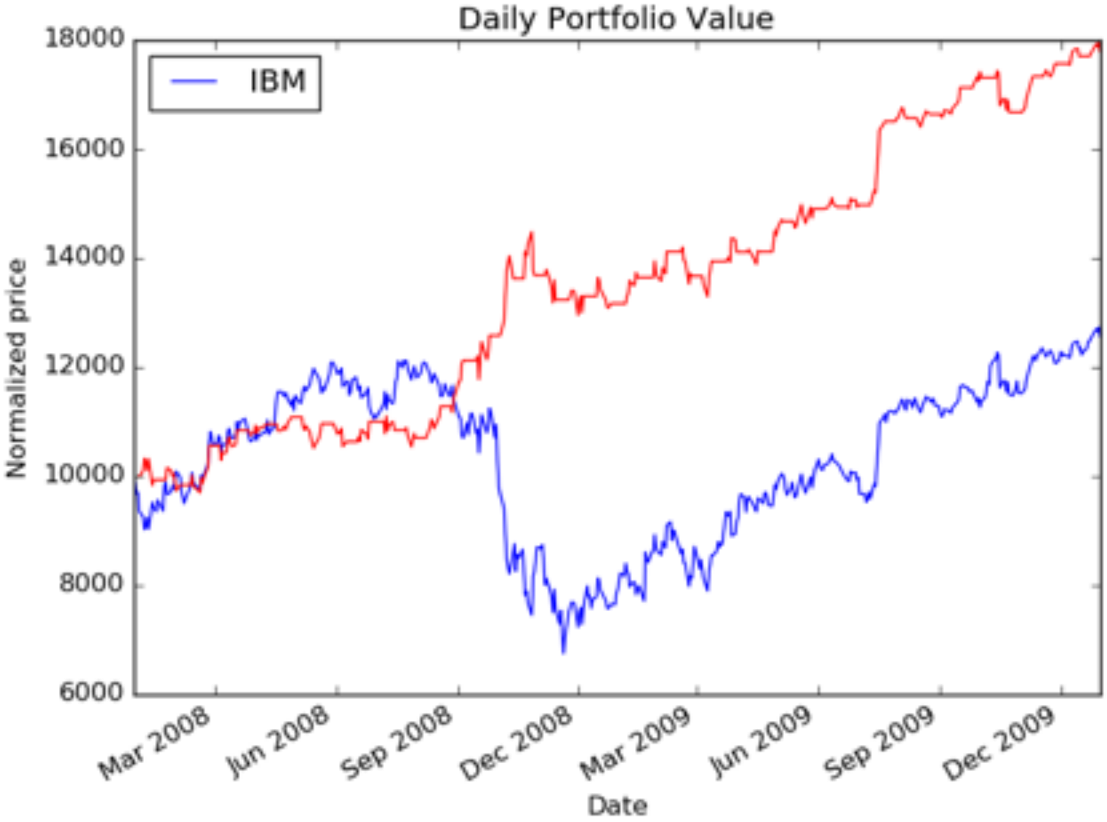
Above is the entry and exit point chart for out of sample ML4T-220. Green lines denote entry into a long position, black lines indicate closing the current position, and red lines denote entry into a short position.



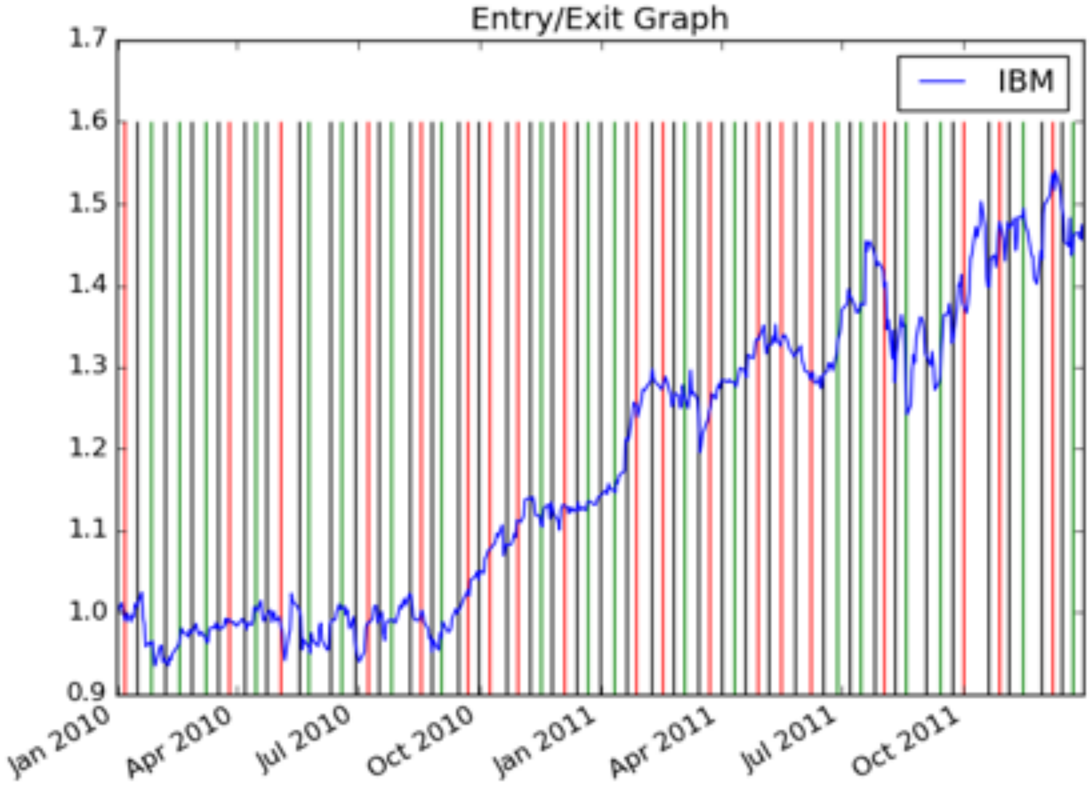
Above is the out of sample backtest for ML4T-220. The learner's performance is in red, while ML4T-220's performance is in blue.



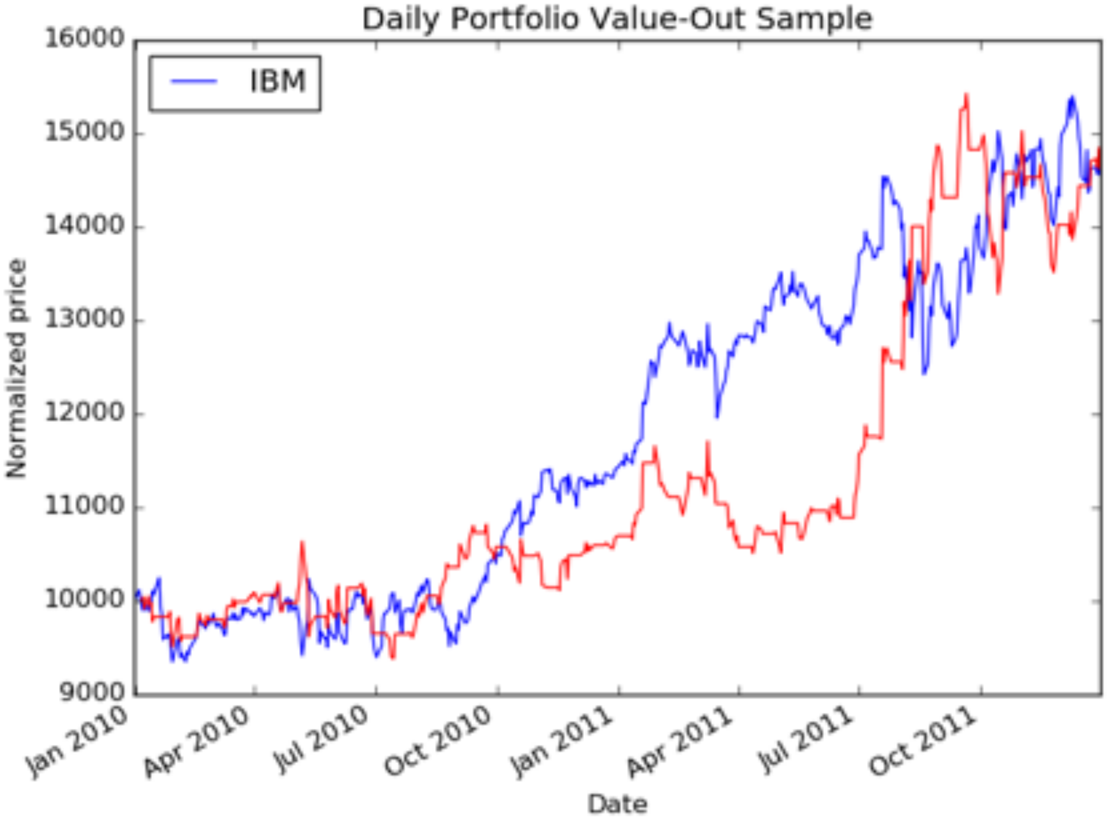
Above is the in sample entry and exit chart of IBM. Green lines denote entry into a long position, black lines indicate closing the current position, and red lines denote entry into a short position.



Above is the in sample portfolio backtest of the learner on IBM. The learner's performance is in **red, while ML4T-220 is in blue.



Above is the out of sample entry and exit chart of IBM. Green lines denote entry into a long position, black lines indicate closing the current position, and red lines denote entry into a short position.



Above is the out of sample portfolio backtest of the learner on IBM. The learner's performance is in **red, while ML4T-220 is in blue.