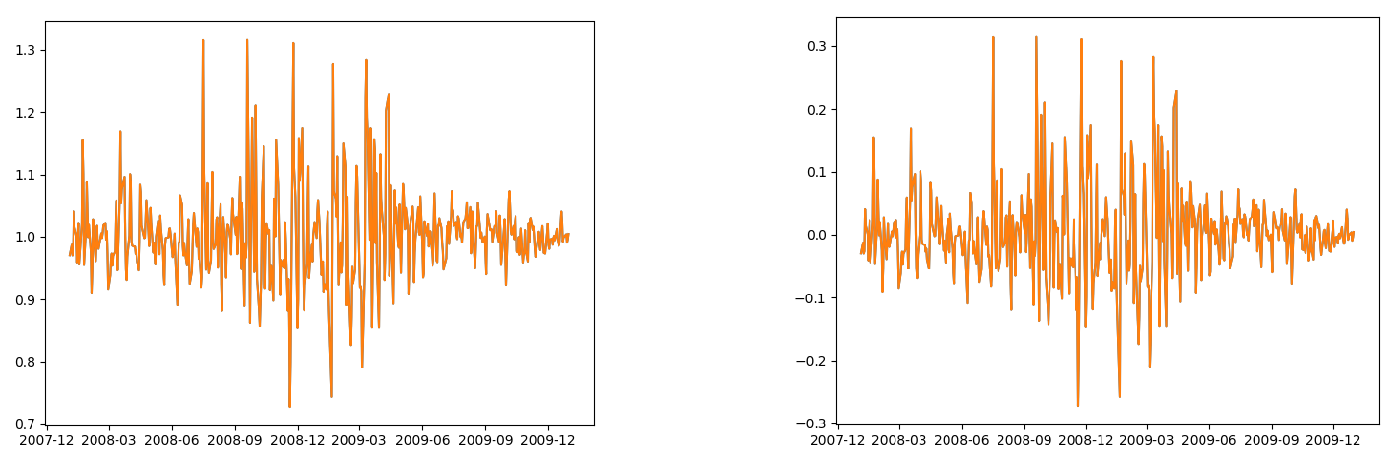
Strategy Learner Docs

I used various machine learning techniques that take historical trends of data, as shown in the indicators.py file, in order to try and accurately predict the price of a given stock. These historical trends come in the form of technical indicators. The machine learning algorithm used here is that of a random forest learner. In addition to this, I also used Bootstrap Aggregation on top of these random forest learners, using 20 bags. The random forest learner picks attributes randomly in this scenario. While this does give us different results each time we run the program, we can see a general trend in the grand scheme of things. The leaf size used was 3. This value was picked via trial and error to give the best results for the Strategy Learner.

Indicators

Momentum:

Momentum trading is a technique in which traders buy and sell according to the strength of recent price trends. It is calculated like this : The difference between the price at a given point and the price N days before that point, divided by the price N days before that point. Once again, via trial and error, N was chosen to be 3.

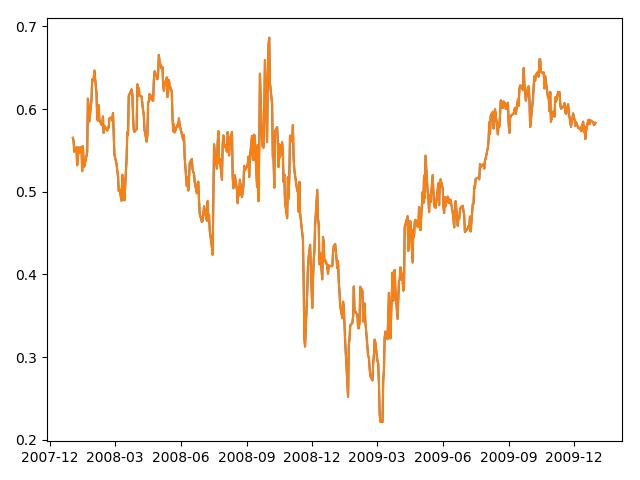


*Figure 1 Momentum graph normalized* *Figure 2 Momentum graph used for decision making*

The graph above on the left shows the normalized momentum graph. The graph used for the actual threshold measurements is the one on the right since we’ve subtracted 1 from it. A value above 0.01 is a buy and one below -0.01 is a sell.

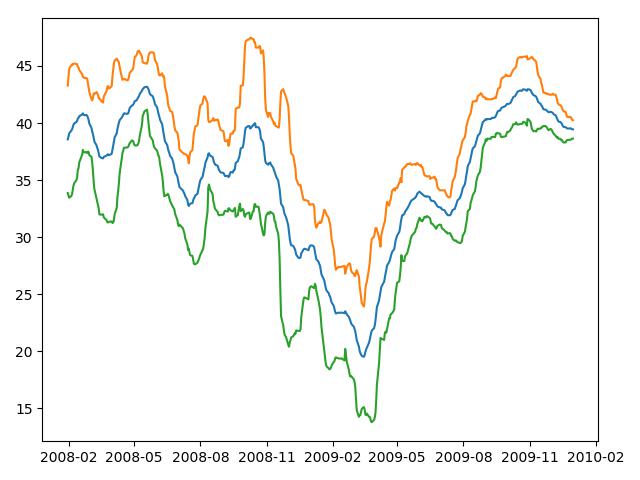
Midpoint:

The midpoint is also calculated using a previous N days as a reference. You have to take the average of the maximum and minimum price as found in the N day period. The prices of each day are divided by their midpoints as an indicator. This is what is shown on the graph below.



*Figure 3 Midpoint graph*

Bollinger Bands: The Bollinger Bands are calculated as shown in class. First we have to find the simple moving average over the past N number of days, which can be set in the code. I chose an N of 20, also from trial and error. The simple moving average, as a reminder, is the average price of the stock over the past N days from the current point in time. As a result, the first 20 days of the data frame is NaN since there isn’t enough data yet for the calculation. The calculation is repeated from the first point where the value is not NaN until the end of the specified period. After this, you have to set the upper and lower bands. The upper and lower bands are calculated using the simple moving average as a reference, plus(upper band) and minus(lower band) two times the standard deviation.

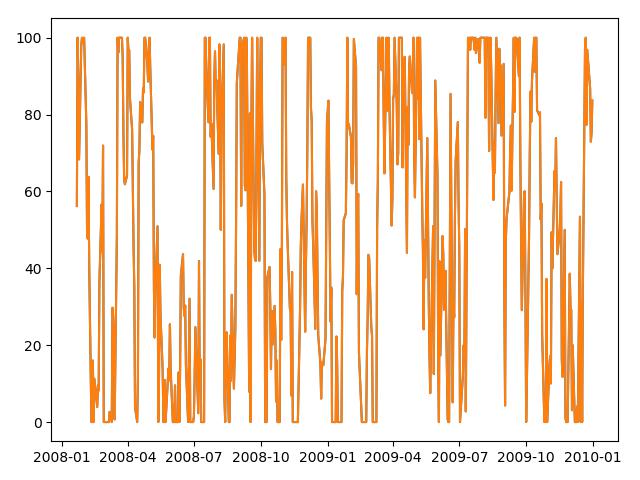


*Figure 4 Bollinger Band Visualization*

Here is a plot of the Bollinger bands. The lower band is represented by the green band and the upper band is represented by the orange line. The blue line shows the simple moving average of the stock, which is JPM in this case.

Stochastic Indicator:

The stochastic oscillator is a momentum indicator comparing the closing price of a security to the range of its prices over a certain period of time. The sensitivity of the oscillator to market movements is reducible by adjusting that time period or by taking a moving average of the result. The range for this is between 0 and 100. If it is below 20, it is considered oversold and it is the best time to buy. Conversely, if it is above 80, it is considered overbought and it is time to sell.



*Figure 5 Stochastic Indicator graph*

The classification (Y Value) determines our decision with the stock. A classification of +1 is given if we should go long, a classification of -1 if we should short it, or simply 0 indicating that no position should be taken at all. How are these numbers determined? I look at the upcoming week’s returns, at 7 days, and make a decision from there. If this return is above a certain threshold, we assign a +1. If they are below a certain threshold, we assign a -1. However, if it is between these two thresholds, a value of 0 is assigned. Once the learner is trained on these features (i.e – indicators), we can use the same thing to maximize the next week’s returns. The threshold was picked as 1% or 0.01 in code. This value is negative for the threshold where we assign a -1.

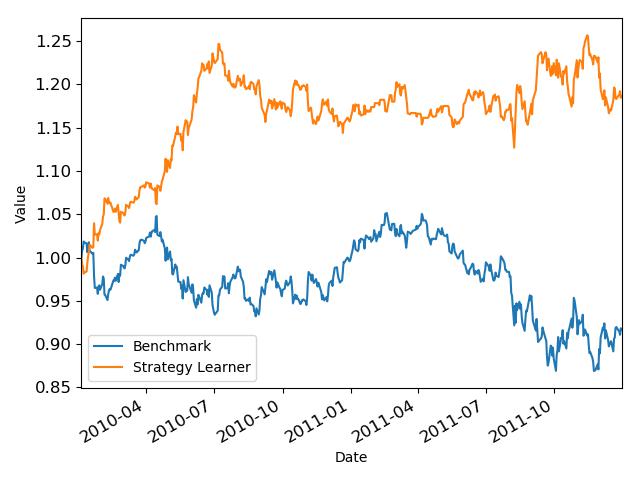
Experiment 1

For experiment 1, the goal was to compare the performance of the strategy learner with the benchmark which was used as a ‘null hypothesis’ of sorts. The machine learning strategy used the same indicators as above (Bollinger bands, momentum and midpoint), as well as used the Random Forest Learner alongside the Bag Learner. The in-sample training, for the symbol JPM, was from January 1st 2008 up to December 31st 2009.

Here is how I act upon my given indicators:

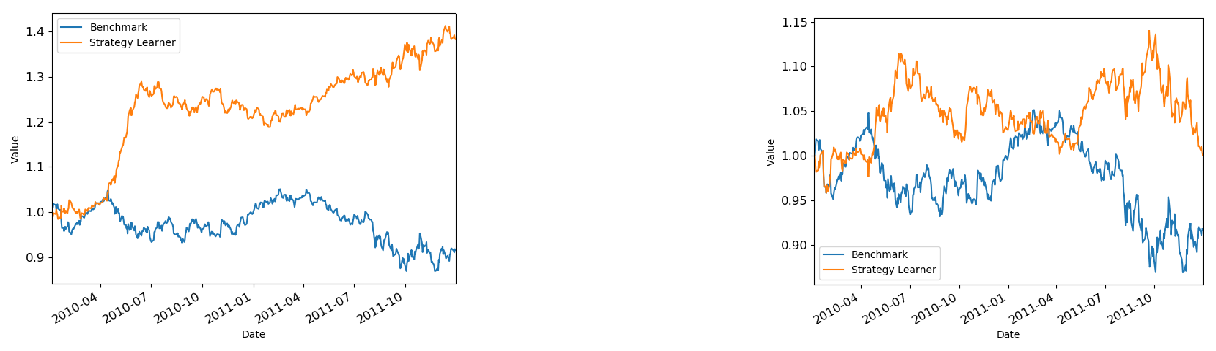
* If the stock price passes up through the lower Bollinger Band after going below it, short the position 1000 shares.
* If you have shorted the position in the past, and the price goes above the midpoint via a positive momentum indicator, close out the position by buying 1000 shares.
* If the stock price passes down through the upper Bollinger Band after going above it, long the position 1000 shares.
* If you have gone long for a given position in the past, and the price goes below the midpoint via a negative momentum indicator, sell 1000 shares.

The benchmark portfolio was used as a basis for our strategy learner. The benchmark is measured by considering the scenario if we simply bought the stock on day one and just held it till the end of the period we are considering. At the very least, we must do better than this. Otherwise, what is the point of even applying all these machine learning techniques? In this experiment, there was no commission or market impact. Here is what the result of my experiment looked like for out of sample data.



*Figure 6 Returns using a 'lookback' period of 7 days*

After this, I decided to try and see what it would be like for something like 5 or 10 days. Here are the results:

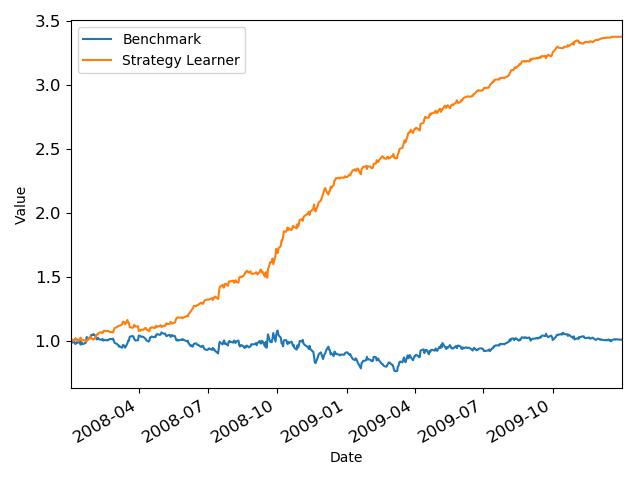


*Figure 7 Returns using a 'lookback' period of 5 days* *Figure 8 Returns using a 'lookback' period of 10 days*

We can see here that having it at 5 days provided the highest returns, although a period of 7 days was definitely a close second.

However, for the in-sample data, we got stunningly more lucrative results. This is to be expected since we are simply telling the model to predict what it has been trained on. This isn’t a problem since there is nothing new or surprising about the indicators provided and it can, in a way, pretty much copy what it has been trained on.

Here is what that looked like:



*Figure 9 Results using In Sample data for testing*

We can see that, if we could somehow go back in time and trade using this knowledge, we could have made a little over triple the returns than usual.