Daniel Mantica
User ID: dmantica3
CS 7646

Strategy Learner Report

The goal of this project was to create a machine learning based trader that could take the historical price data of a stock and technical indicators calculated from that price data to create a profitable trading strategy. The machine learning algorithm I chose to use was a Random Forest classification-based learner. The Random Forest learner is built using the method of Bootstrap Aggregating, where a set of Random Tree learners is aggregated to create one model. For my Random Forest learner, I used 10 “bags” (10 Random Tree learners combined into the Random Forest). Each Random Tree learner is built by randomly determining different factors and values on which to split the data into parts until the number of data observations is less than or equal to predetermined “leaf size”. For my Random Tree learners I used a leaf size of 5. The factors that are used to split the data are a set of technical indicators calculated from the price of the stock. These are the technical indicators I used:

1. Bollinger Bands: The values of the Bollinger Bands of the stock price at each point in time are calculated by first calculating the average price of the stock over the past N days from that point, referred to as the lookback period. I used a lookback period of 20 days (N = 20). This average price is called the simple moving average. Next, the standard deviation of the stock price during the lookback period is calculated. Finally, the upper Bollinger Band value is calculated by taking the simple moving average plus two times the standard deviation, while the lower Bollinger Band value is calculated by taking the simple moving average minus two times the standard deviation. This calculation is repeated for each day of the time period, starting N days after the start of the period (since this is the point after which the simple moving average can start to be calculated). The four separate indicator values are then calculated by taking both the current day’s price and the previous day’s price divided by the upper and lower Bollinger Band values for each day of the period.

2. Momentum: Momentum describes the relative change in the stock price over a certain period. It is calculated at each point in time by taking the price at that point minus the price N days prior, then dividing the result by the price N days prior. To calculate momentum I used N = 3. The momentum indicator value is then simply taken as the momentum value.

3. Midpoint: The midpoint at each point in time is calculated by once again using a certain lookback period, just as used to calculate Bollinger Bands, but instead of using the average price of the stock, it uses the average of the maximum and minimum price of the stock during the lookback period. For calculating the midpoint I used a lookback period of 100 days. The two separate midpoint

indicator values are then calculated by taking both the current day's price and the previous day's price divided by the midpoint values for each day of the period.

I used the ratio of the price divided by the indicator for the Bollinger Bands and midpoint indicators in order to standardize the values of the indicators. This standardization ensures that each indicator contributes proportionally to the learning algorithm.

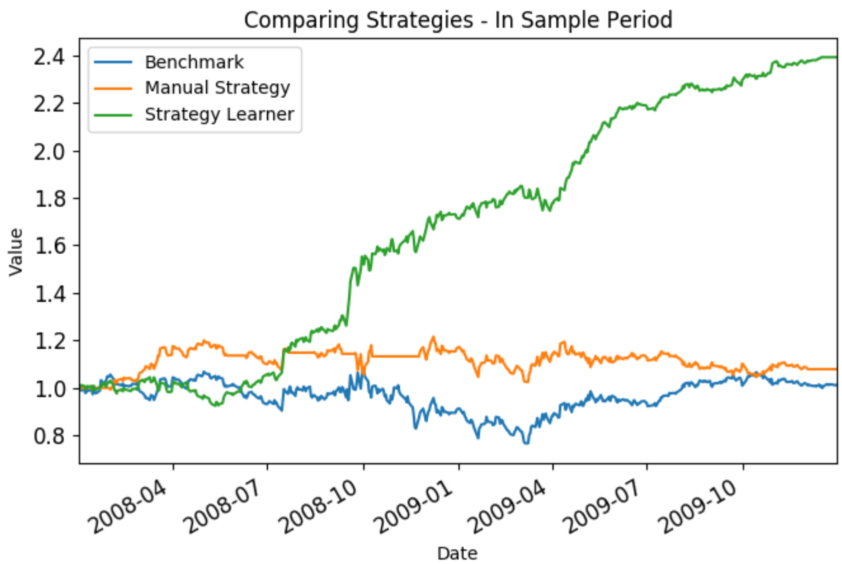
The data used to train each of the Random Tree learners of the Random Forest learner is comprised of a set of each of the technical indicators calculated for each day of the in-sample training period. In addition, each data set is assigned a classification of either +1, indicating that a long position on the stock should be taken, -1, indicating that a short position should be taken, or 0, indicating that no position should be taken or any currently held positions should be closed out. This classification value is determined by looking at the future returns of the stock some days ahead in the future. For this report I chose to use the 10-day return. If the 10-day returns are above a certain threshold YBUY plus the market impact of a trade, a +1 classification is assigned; if they are below a different threshold YSELL minus the market impact, a -1 classification is assigned; if they are neither above YBUY plus market impact nor below YSELL minus market impact, a 0 classification is assigned. I chose 0.01 for the value of YBUY and -0.01 for the value of YSELL. Once the learner is trained, it can be given a set of the technical indicators for a stock calculated on a given day, and it will then determine which position on the stock (long, short, or none) should be taken to maximize future 10-day returns.

Experiment 1

For this experiment, the goal was to compare the performance of the trading strategy of the machine learning based trader with that of a manual trading strategy developed in a previous project, trading the symbol JPM. Each strategy used the same technical indicators (Bollinger Bands, momentum, and midpoints) described above. The machine learning based strategy used was created using the Random Forest classification learner described above, using the same parameters (10 bags, leaf size of 5, 10-day returns, YBUY value of 0.01, YSELL value of -0.01). The Random Forest learner was trained using data from an in-sample period of January 1, 2008 to December 31, 2009. The manual trading strategy used the following logic:

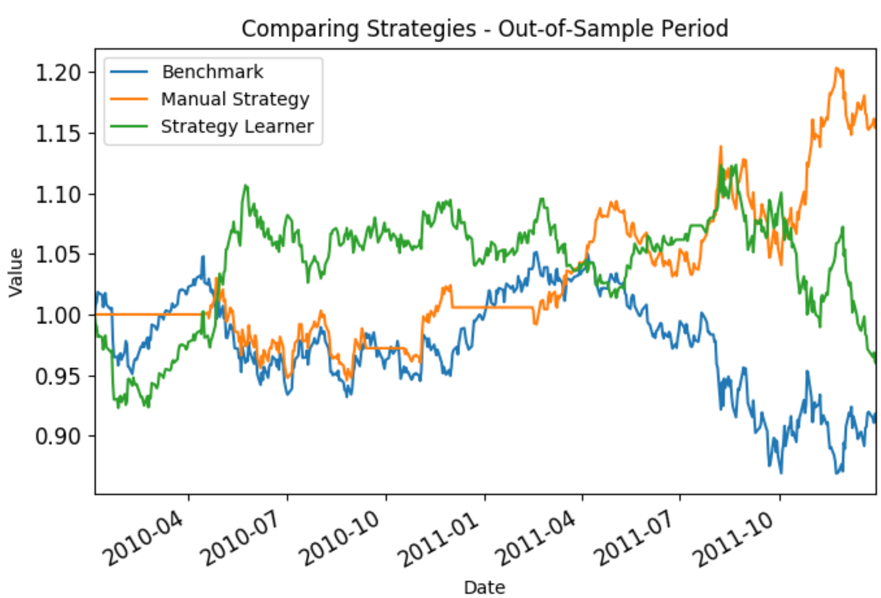
1. If the stock price passes back down through the upper Bollinger Band, make a trade that makes your position short 1000 shares
2. If the stock price passes back up through the lower Bollinger Band, make a trade that makes your position long 1000 shares
3. If you have a long position and the price passes down through the midpoint with a negative momentum indicator, close out the position by selling 1000 shares
4. If you have a short position and the price passes up through the midpoint with a positive momentum indicator, close out the position by buying 1000 shares

In addition, each strategy was compared with a benchmark portfolio, which consisted of buying 1000 shares of JPM on the first day and then holding them throughout the period. For each strategy it was assumed that there were no commission fees and no market impact caused by trades. The only positions allowed were neutral, long 1000 shares of JPM, and short 1000 shares of JPM. Here are the results of the performance of each strategy for this in-sample period, compared with the benchmark:



As expected, the machine learning based strategy outperformed the manual strategy and the benchmark in terms of cumulative return during the in-sample period. This result should be expected every time for in-sample data because machine learning algorithm is trained using the data from this period to maximize returns, while the manual strategy was simply developed by looking at the price data and

Here are the results of the performance of each strategy for the out-of-sample testing period of January 1, 2010 to December 31, 2011, compared with the benchmark:



Now, the machine learning based strategy has a lower cumulative return than the manual trading strategy, though it still outperforms the benchmark portfolio. Although disappointing, this result is not unexpected, because it is expected that the machine learning algorithm will not perform as well when given data that it was not trained on.

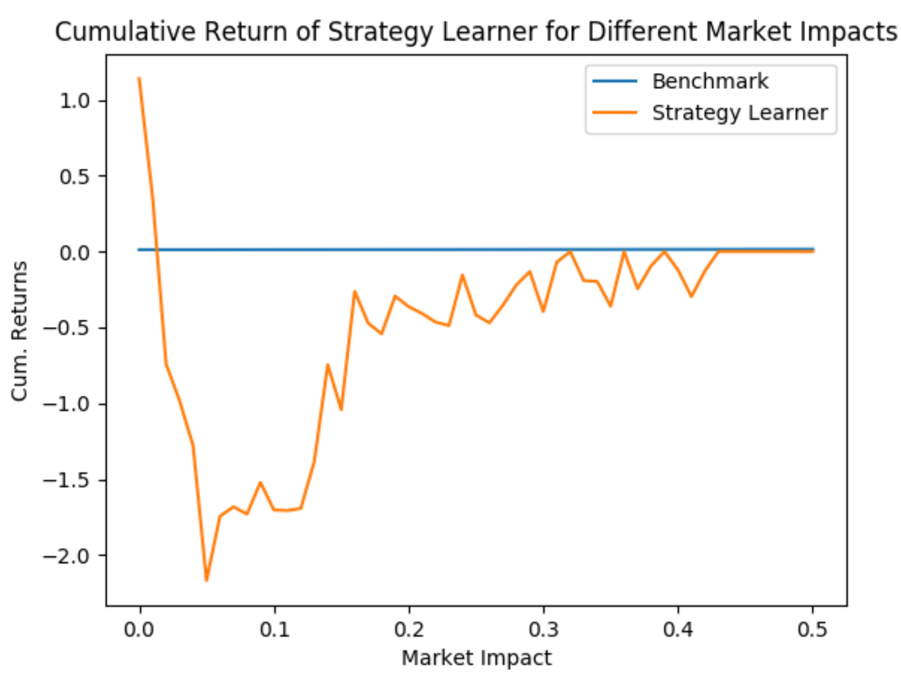
Experiment 2

For this experiment, the goal was to determine what affect changing the value of the market impact of each trade would have on the behavior and performance of the machine learning based trading strategy, developed from the Random Forest classification learner previously described (same technical indicators, 10 bags, leaf size of 5, 10-day returns, YBUY value of 0.01, YSELL value of -0.01), during the in-sample training period of January 1, 2008 to December 31, 2009. I developed two hypotheses for what affect changing the market impact would have:

1. Increasing the market impact will lead to worse cumulative returns for the machine learning based trading strategy.

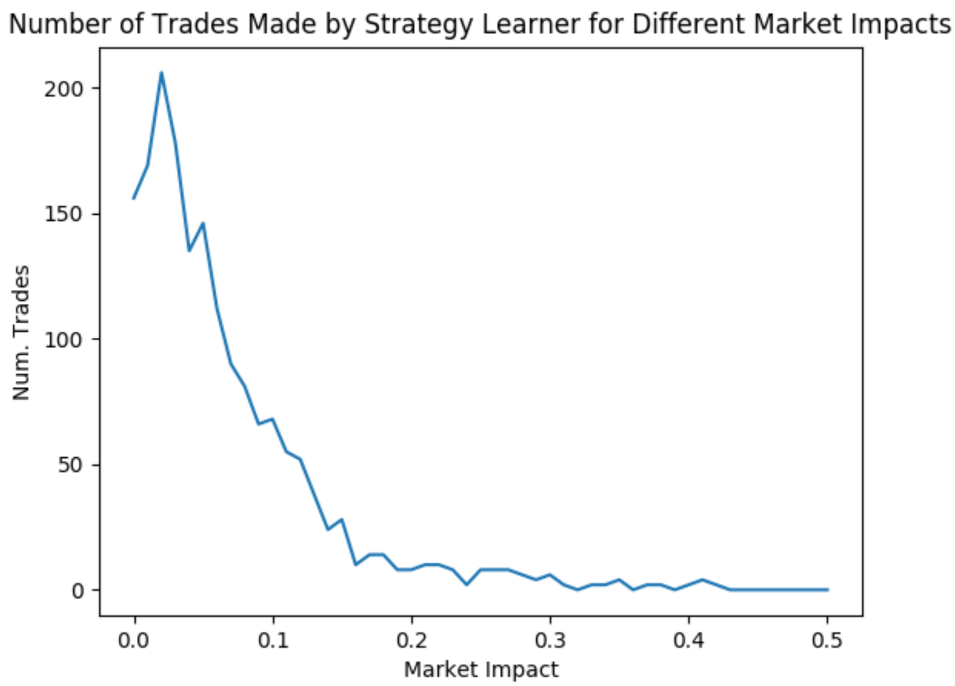
2. Increasing the market impact will lead to the machine learning based strategy making fewer overall trades during the period.

To test these hypotheses, I created Random Forest learners for market impact values ranging from 0 to 0.5, incremented by 0.01, and then calculated the cumulative return of and number of trades made by the trading strategy created by each learner. Here are the results for how changing the market impact affected the cumulative returns of the machine learning based strategy:



At first, the cumulative returns of the learner strategy behaved as expected, decreasing as the market impact was increased. However, at a market impact of 0.06 they began to increase, eventually approaching a cumulative return of 0, which goes against my initial hypothesis. I believe that this increase occurs because as the impact becomes greater, the number of trades made by the learner strategy decreases, so there are less chances to incur losses.

Here are the results for how changing the market impact affected the number of trades made by the machine based trading strategy:



As hypothesized, after the market impact hit 0.02 the number of trades made by the machine learning strategy began to decrease, eventually hitting 0.