Part 1: Technical Indicators

Three technical indicators were selected for use in informing both a manual trading strategy and a decision tree learning strategy. These indicators are price / simple moving average (SMA) ratio, Bollinger Band percentage, and Aroon oscillator.

Computing price / SMA ratio begins with the computation of an *n*-day rolling average. For example, for a 10-day SMA, each day's value is the mean of that day's closing price and the previous nine days' closing prices. With SMA computed, the equity's price on each day is divided by its SMA on that day to determine price / SMA ratio, which normalizes the price to its relationship with the SMA. When the ratio is greater than one, the price is above its SMA, suggesting a potential sell signal, and when it is less than one, the price has dipped below its SMA, which may be a buy signal.

Bollinger Band computation also begins with computing SMA, and then expands upon this by determining the standard deviation of the equity's price over the previous *n* days. Two standard deviations are added to the SMA to determine the day's upper Bollinger Band, and two standard deviations are subtracted from the SMA to form the lower Band. From the Bollinger Bands, a single indicator called Bollinger Band Percentage can be calculated by creating a percentage range between the top and bottom bands, and returning the price's location between the bands in terms of this percentage range. Values greater than 100 percent and less than 0 percent indicate that the price has crossed a Bollinger Band.

Finally, the Aroon oscillator is a measure of price momentum which may signal bullish or bearish trends in an equity's price movement. The oscillator's calculation begins with the computation of two Aroon indicators, one for upward movement and one for downward:

$$Aroon_{up} = 100 * \frac{n - (days\ since\ price\ high\ in\ window)}{n}$$

$$Aroon_{down} = 100 * \frac{n - (days\ since\ price\ low\ in\ window)}{n}$$

Finally, the Aroon oscillator value is calculated by determining the difference between Aroon_{up} and $\text{Aroon}_{\text{down}}$; this sets the range of the oscillator between 100 and -100. Values above 70 are often interpreted as signals of significant upward momentum, while values below -70 are interpreted to signal significant downward momentum.

To enable meaningful weighting and consideration of each indicator during the decision tree learning process, the indicators were standardized into z-scores by using the following conversion:

$$z = \frac{x - \mu}{\sigma}$$

where μ represents the mean of the indicator scores and σ represents the standard deviation of the scores. The z-score represents the distance between the raw value and the mean of the values, in multiples of standard deviations. An indicator value which is two standard deviations above the indicator's mean would therefore receive a z-score of 2.0.

Each of the indicators within the in-sample period are shown below along with the raw price data.

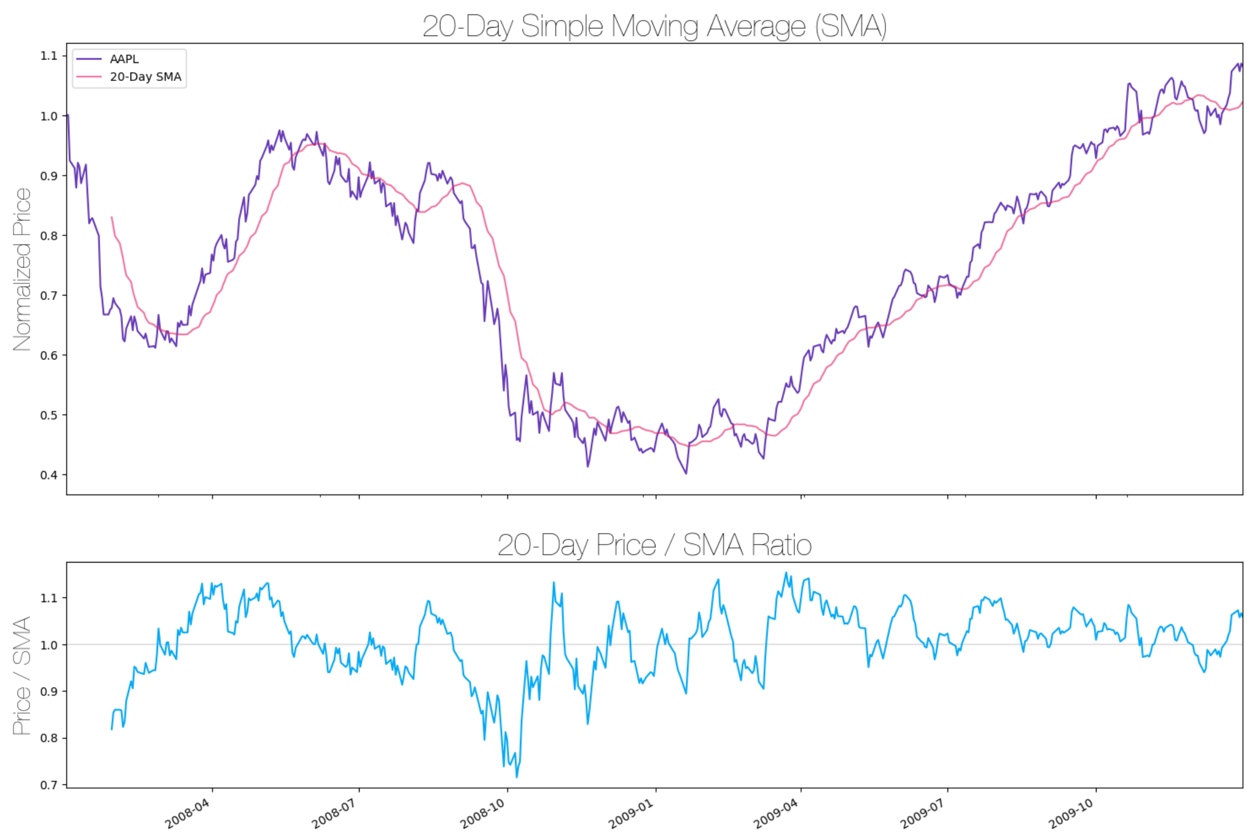

Figure 1- Normalized 20-day SMA and price / SMA ratio for AAPL within the in-sample period.

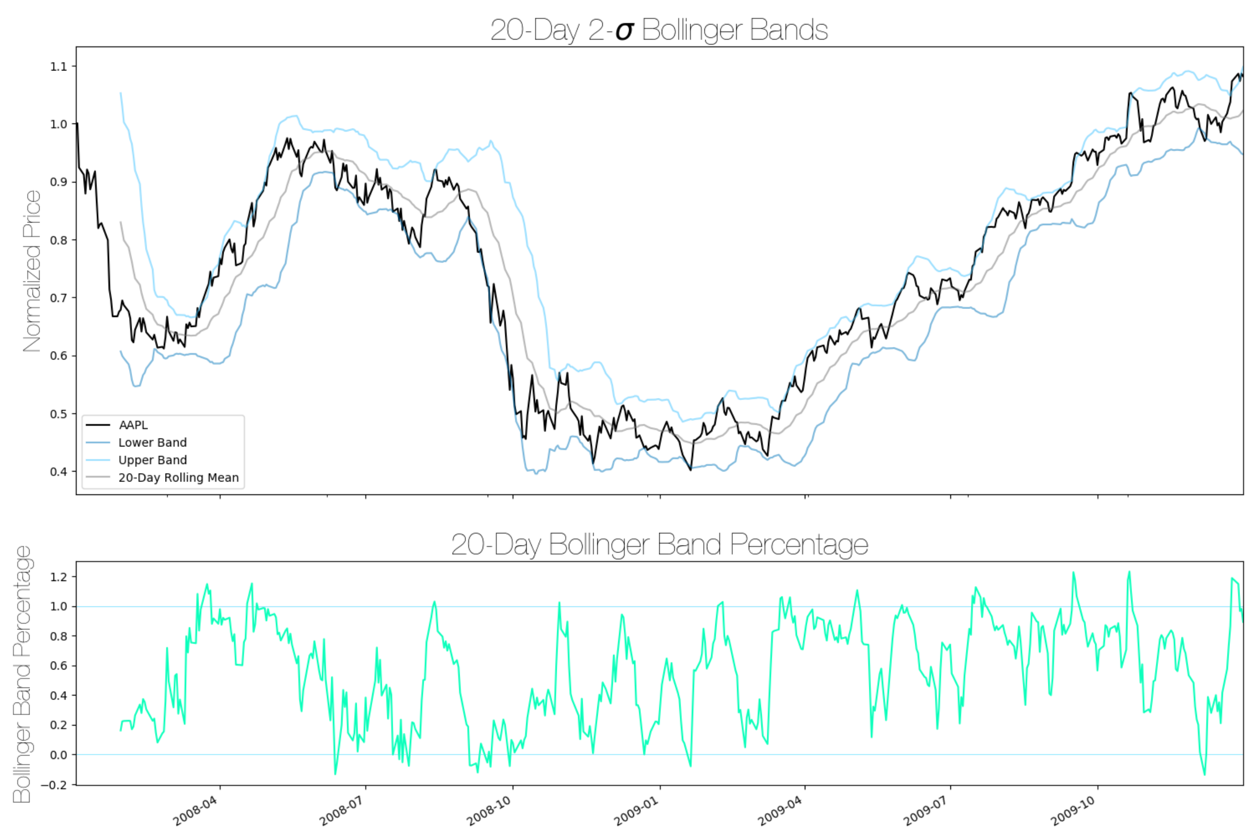

Figure 2- Bollinger Bands and Bollinger Band Percentage for normalized price data for AAPL within the in-sample period.

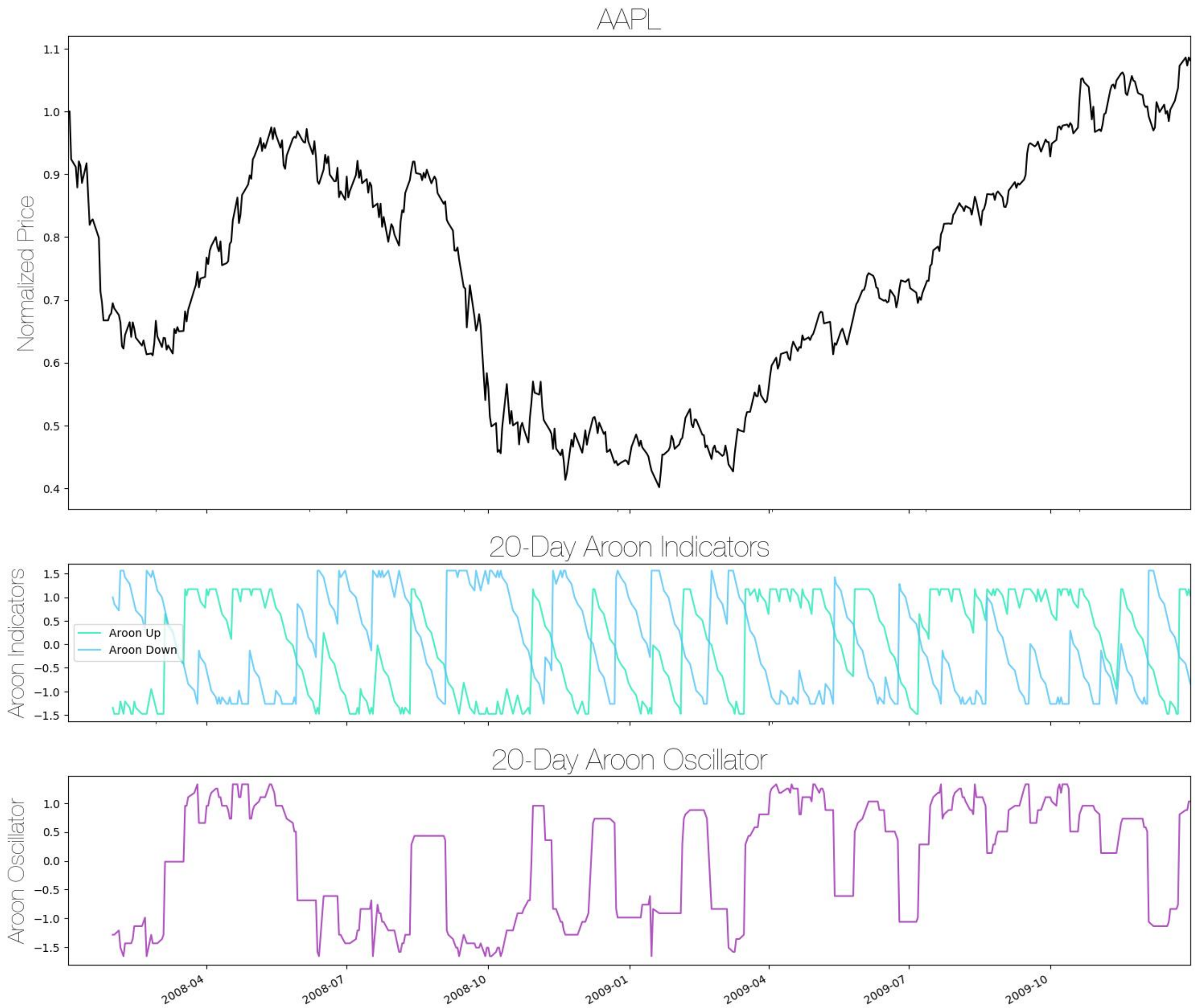

Figure 3-Standardized Aroon indicators and Aroon oscillator for a 20-day window of normalized AAPL price data within the in-sample period.

Part 2: Best Possible Strategy

The best possible strategy for the chosen equity in the in-sample training period was defined as that strategy which buys 200 shares when the following day's closing price is greater than the current day's, and which sells 200 shares when the following day's closing price is lesser than the current day's. Using this strategy, a set of orders was generated; the returns of this best-possible strategy are shown below, compared against a benchmark strategy in which 200 shares were purchased on the first day of the period and held thereafter.

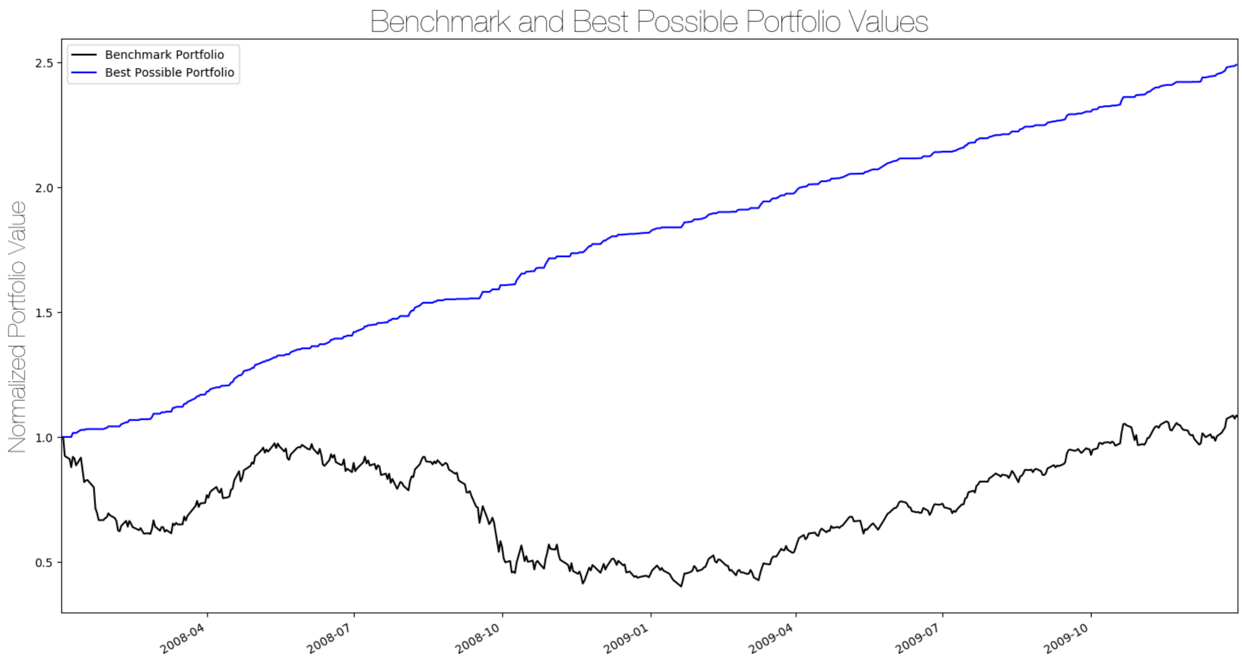

Figure 4- Normalized daily values of benchmark and best possible portfolios

Portfolio statistics were printed for each of these two portfolios:

```
Cumulative Return of Benchmark: 0.081542188547
Cumulative Return of Best Possible Strategy: 1.49056
Standard Deviation of Benchmark: 0.0301868507055
Standard Deviation of Best Possible Strategy: 0.00282175520461
Average Daily Return of Benchmark: 0.00061363496698
Average Daily Return of Best Possible Strategy: 0.00181972778926
```

Of note is that the best possible strategy here assumes perfect information regarding the adjusted closing price of AAPL one day in the future.

Part 3: Manual Rule-Based Trader

Using the indicators described above, a set of rules for trading a security was devised with the constraints that the only allowable positions are 200 shares long and 200 shares short and that each position must have a minimum duration of 21 trading days.

The strategy employed was:

- Go long when equity is oversold
- Go short when equity is overbought

The equity is classifed as overbought when:

- Price / SMA ratio > 1.02
- Bollinger Band percentage > 0.7
- Aroon oscillator < -0.5

The equity is classified as oversold when:

- Price / SMA ratio < 0.98
- Bollinger Band percentage < 0.4
- Aroon oscillator > 0.5

Using this strategy, a portfolio was generated and then compared against the benchmark strategy. The results are shown below, and demonstrate greater return using the manual strategy:

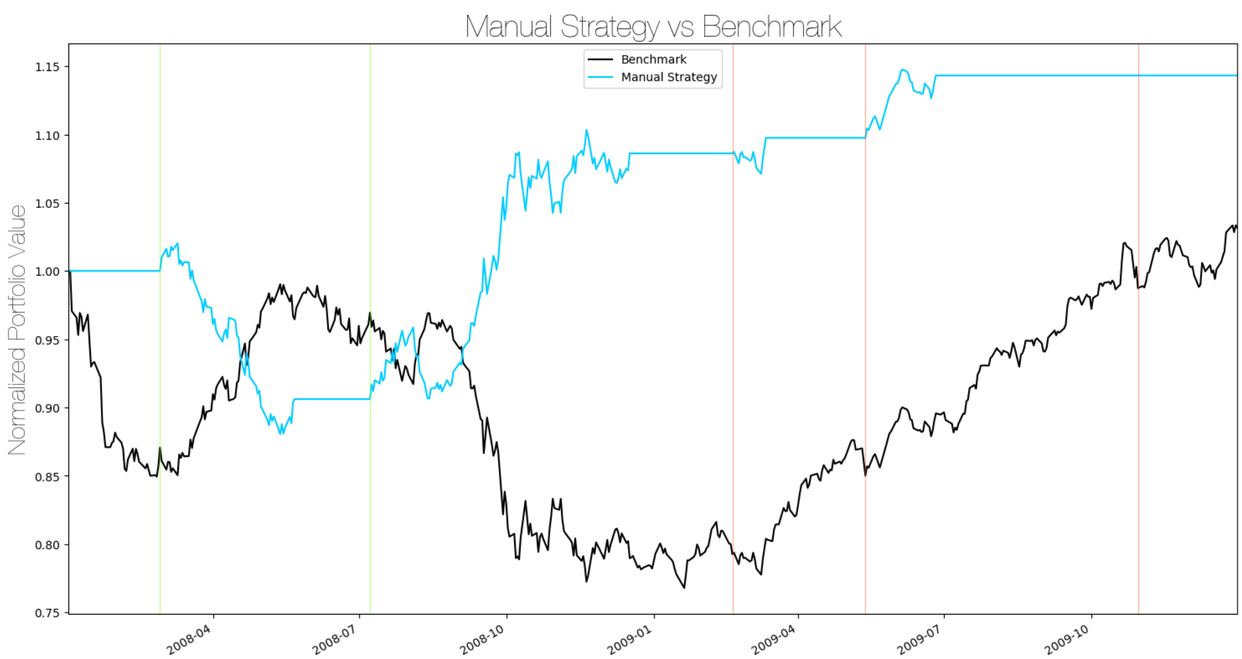

Figure 5- Normalized portfolio values for the benchmark portfolio and for a portfolio built using the manual rule strategy described above. Short position entry points are marked with green lines; long position entries are marked with red lines.

Part 4: Machine Learning Trader

A bagged random tree learner was employed in an effort to learn the predictive value of each of the indicators determined above. The learner used was a random decision tree learner, which splits random features of the data; this particular decision tree learner was modified to provide classification, as opposed to regression. The training data used to build the decision tree was 21-day equity returns, which were classified into y values of -1, 0, or 1, where values less than `short_at` were classified as -1, values greater than `long_at` were classified as 1, and all others were classified as 0. By building the decision tree with these y values, and then rounding the average of the values in the result node when querying the tree, the random tree learner produced values of -1, 0 and 1 for standardized indicator value inputs.

Outputs from the decision tree learner were then used to build an orders file, where outputs of 1 represented BUY, outputs of -1 represented SELL, and outputs of 0 indicated DO NOTHING. The orders file was then modified for compliance with minimum hold time and position constraints.

The `long_at` and `short_at` values used for classifying the y values were modified in order to maximize returns of the machine learning trader. Additionally, multiple random tree learners were bagged together to create a bagged learner. Finally, the leaf size used to build each decision tree in the bagged learner was varied to maximize portfolio returns with the machine learning trader. Peak returns were obtained using 50 bagged random tree learners with a leaf size of 6, a `short_at` value of -0.005, and a `long_at` value of 0.005. The returns of this portfolio are shown below:

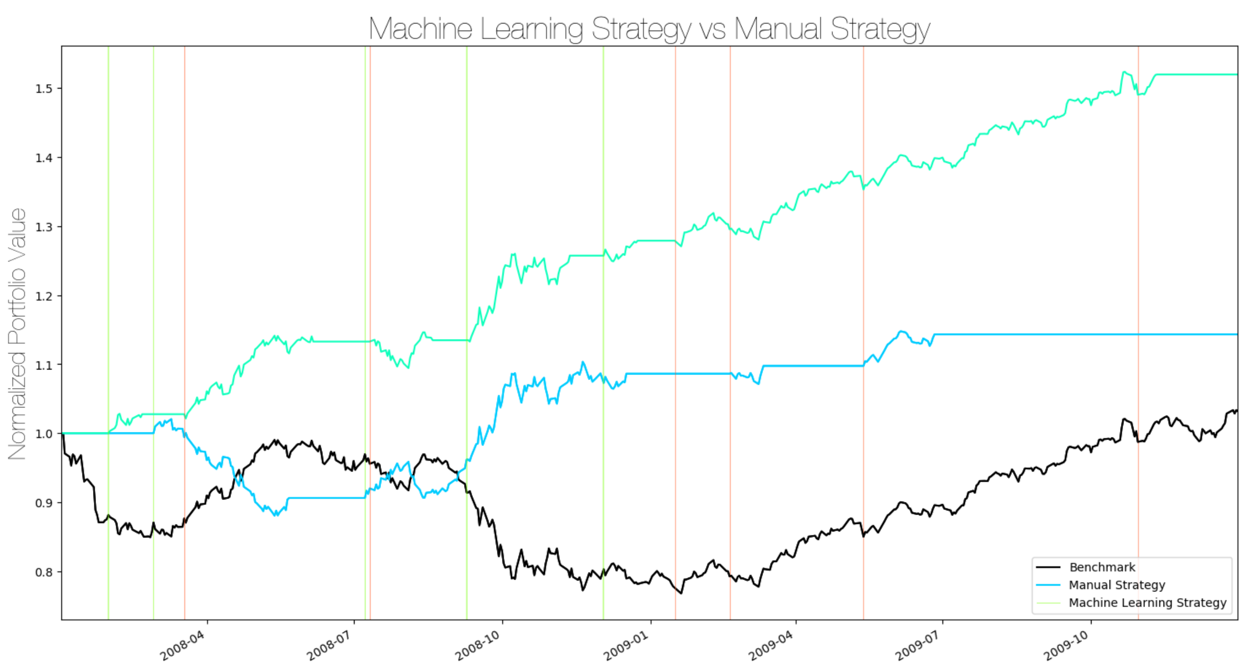

Figure 6- Normalized portfolio values for the benchmark strategy, the manual rule strategy, and the machine learning strategy. Green vertical lines denote short position openings; red vertical lines denote long position openings.

The machine learning trader achieved returns that were 1.5 + the returns of the benchmark strategy, indicating success in producing a functional decision-tree-based trader.

Part 5: Data Visualization

To explore the decision space of the training data set and how it was classified by each strategy, each data point was plotted by its standardized price / SMA and Aroon oscillator indicators values representing the x and y axes respectively, and colored according to how the point was classified by each strategy. The results are shown below:

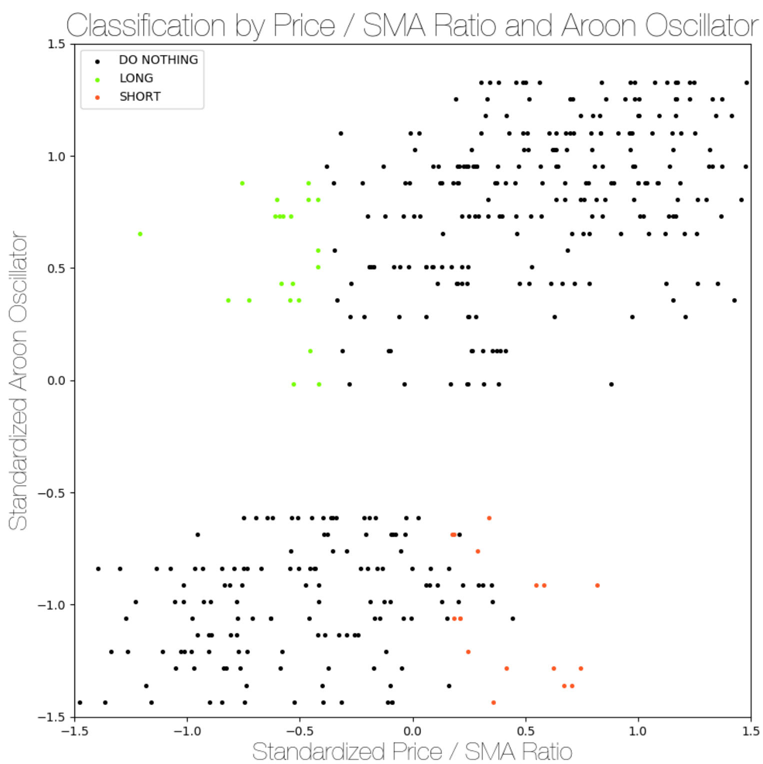

Figure 7- Decision space for manual rule-based strategy

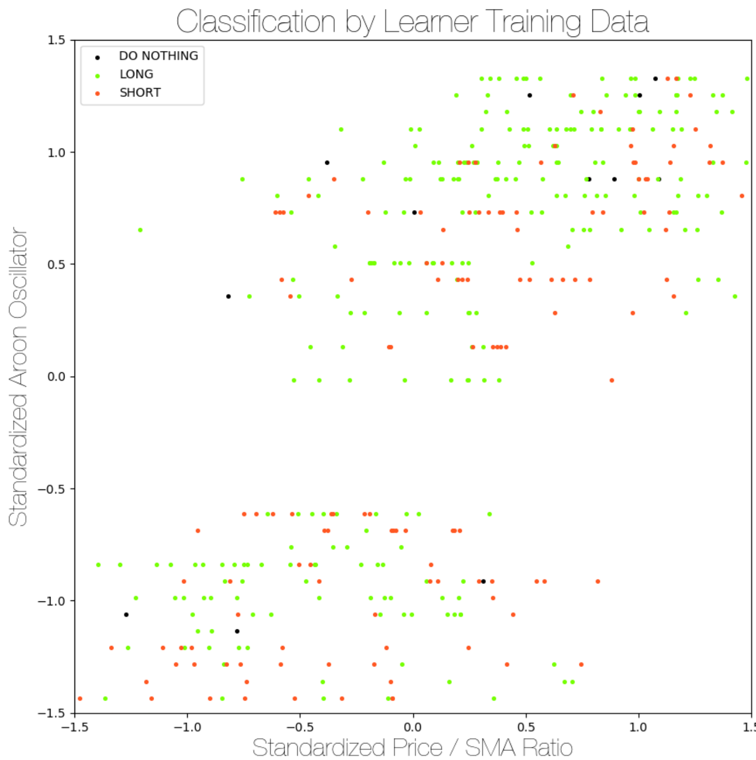

Figure 9- Decision space for the training data set used to train the bagged learner, generated using the criteria described above

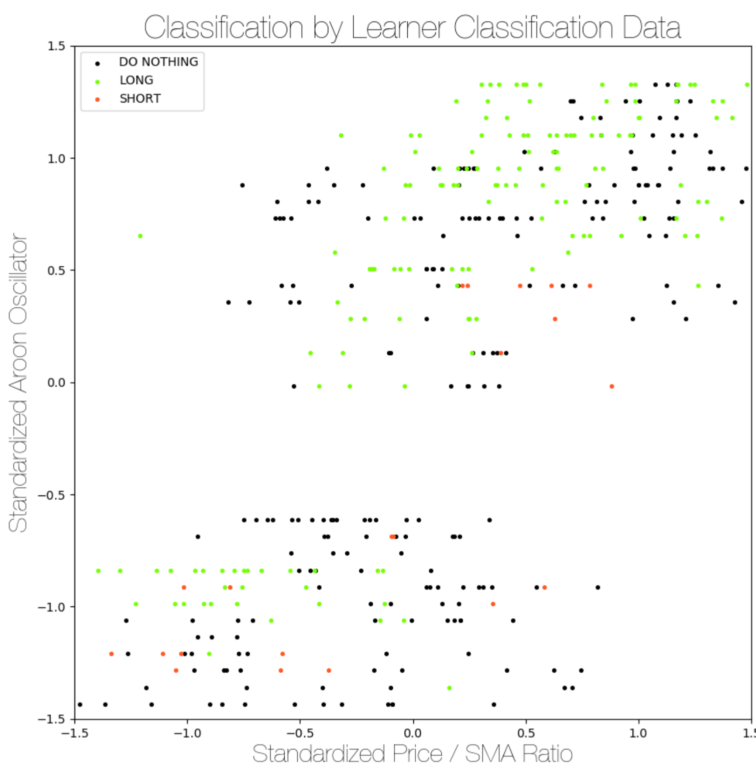

Figure 8- Decision space as generated by the bagged learner

Part 6: Comparative Analysis

With the decision tree learner trained and modified to deliver high in-sample performance, both the manual rule-based strategy and the learner were compared against the benchmark strategy for out of sample data. The results are shown below, and show that the learner trader performed best, followed by the benchmark strategy, and the manual strategy performed the worst. This is consistent with the idea that the manual rule-based strategy is too general to provide meaningful predictive power for the out of sample test data; selecting single threshold values for all market regimes may not be a successful strategy when perfect information of future pricing is unavailable. By contrast, the learner strategy allows the learner to map a finite number of samples directly to their associated prediction, allowing the learner to generalize much better to future price data.

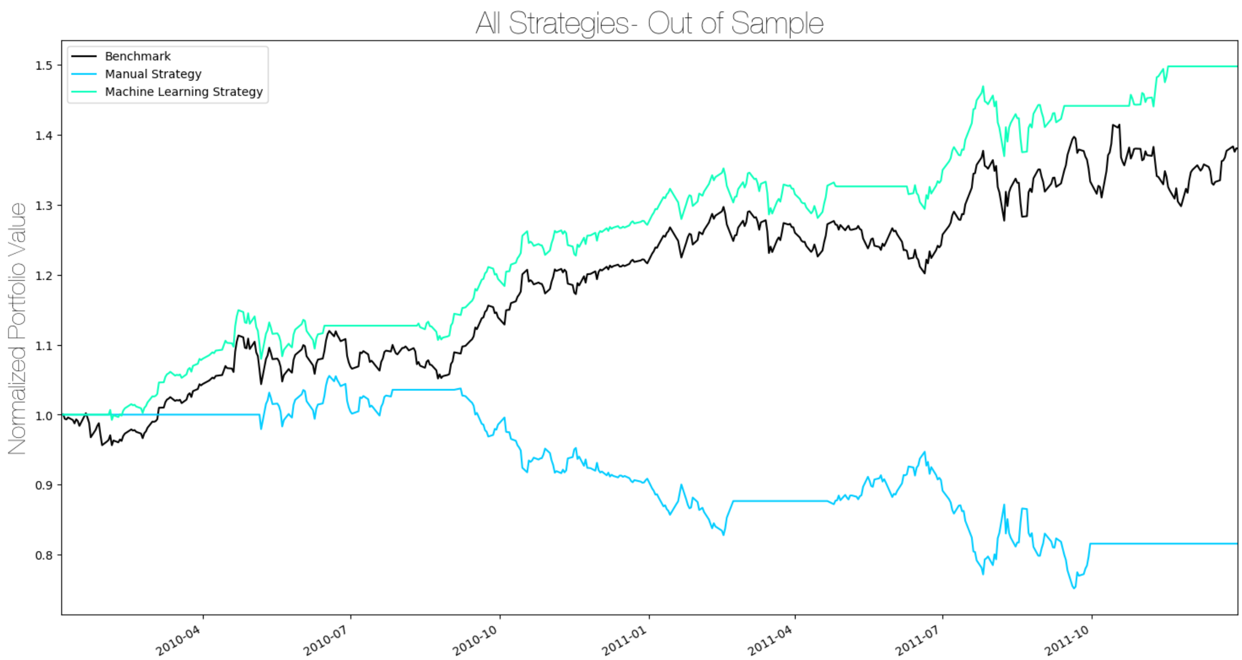

Figure 10- Benchmark, manual, and learner strategy performance on out of sample test data

	Cumulative Return	Standard Deviation	Mean Daily Return
Benchmark	0.3803	0.0086	0.00068
Manual Rule Strategy	-0.1845	0.0086	-0.00034
Bagged Tree Learner	0.4978	0.0066	0.00081

The tabulated portfolio statistics above show that the learner strategy outperformed the other strategies in all metrics, delivering greater average and daily returns with a more favorable risk profile. These results show that even a simple machine learning trader, if well built, can outperform both buy-and-hold trading strategies and actively managed portfolios using the rule-based strategy described here.