# Predicting Bitcoin:
Time Stamped Results

Summary

TradeBot is software made of two parts: a prediction engine, based on an ensemble of regressions, and a trading engine, which analyzes those predictions to find optimal points to buy and sell. Since February 11, TradeBot has issued predictions in real time every 15 minutes, and time stamped those predictions every day.

This paper will demonstrate first the effectiveness of TradeBot's predictions, based on the ability to anticipate the shape of the future price curve, and then show the results of its trading.

Prediction Moving Average—Drawing the Price Curve

Below is the 20 day moving average for the full set of raw predictions, with predictions normalized to price:

Price vs Predictions Moving Average, Feb 11 - Aug 17
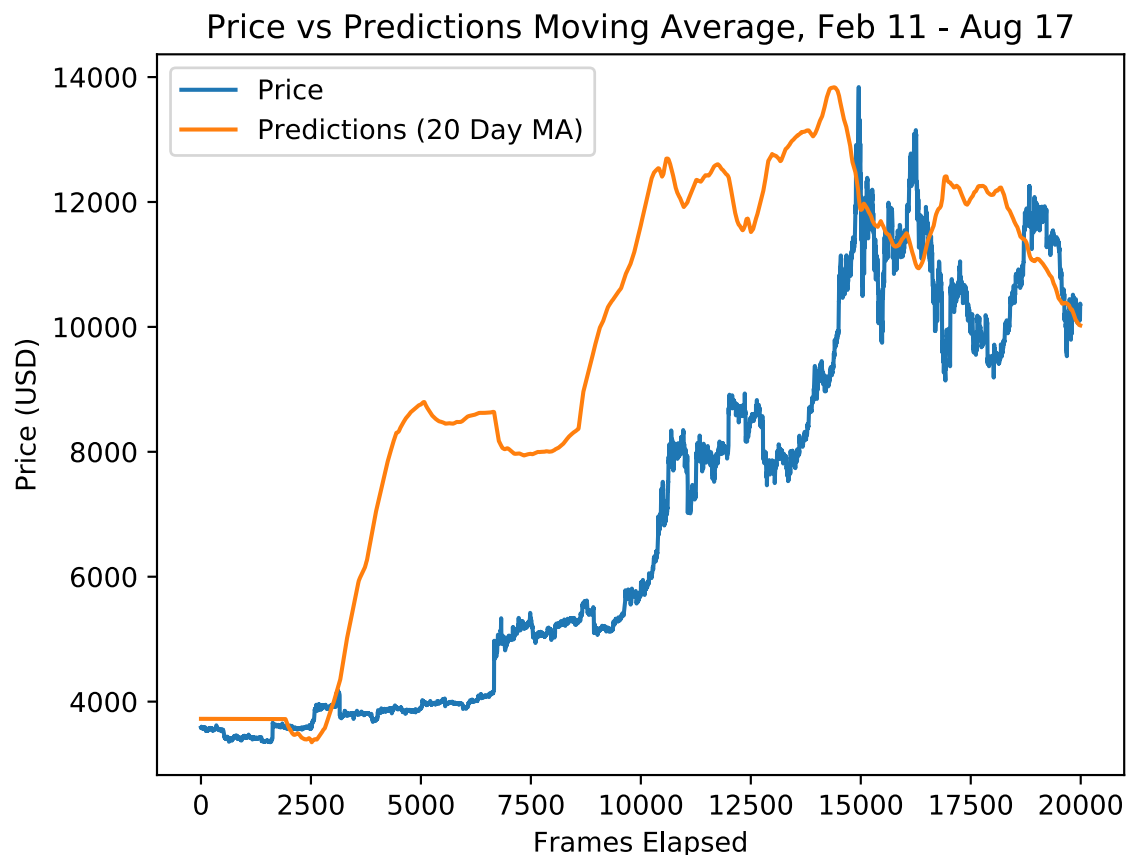
What we're looking for here as an indication of success is whether the direction of the moving average anticipates changes in the direction of price; the clear similarity between the curves is a good indicator that it does. The moving average peaks shortly before the first peak of the price, and begins rising before the second peak; it makes new highs before Bitcoin's price does the same and hits 13,800.

Although the 20 day was extremely accurate on the way up, on the way down it was not quite as accurate—or at least, it's not as visible to the naked eye. Here's the 30 day:
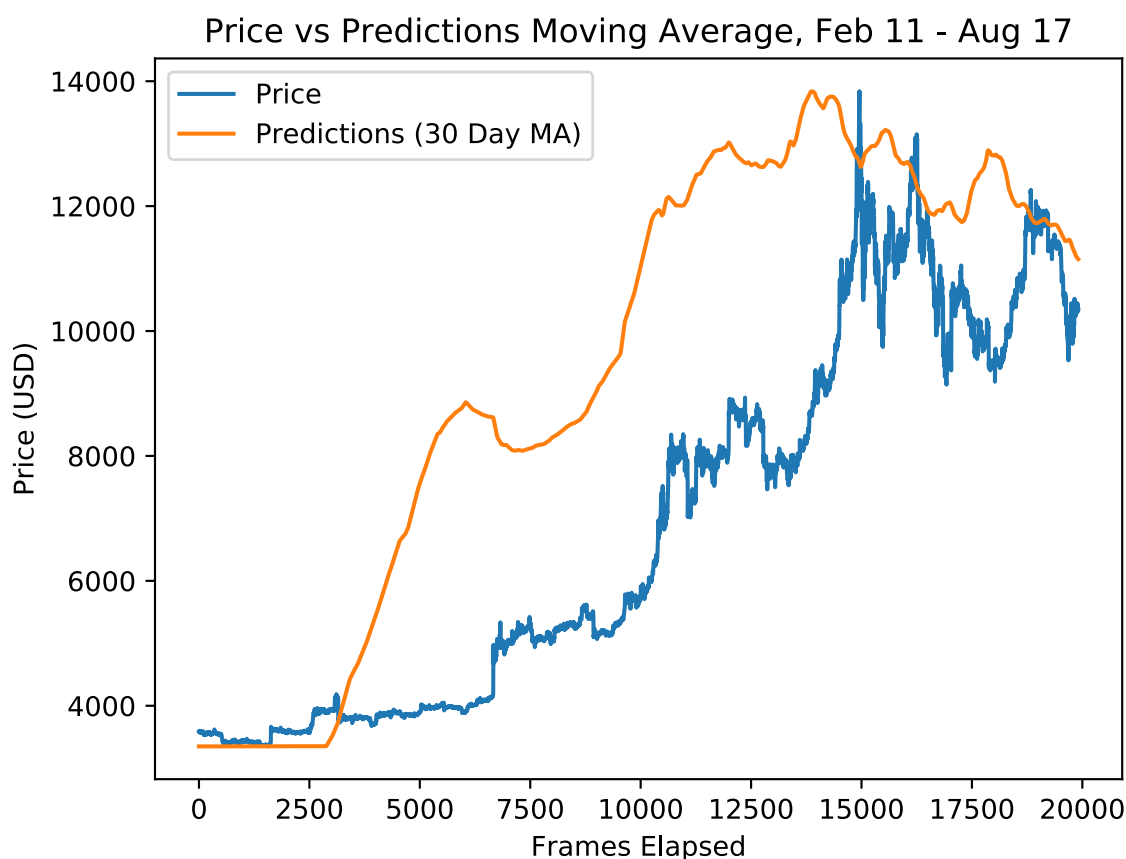

The 30 day, though less descriptive than the 20 on the way up, paints the structure more accurately on the way down. Three peaks of declining size, followed by a fourth and another decline—and all painted in advance of the actual price movement.

Clearly, our predictions are translating some signal of the future—and for six months, real time predictions in a moving average from 20 to 30 days long have painted the price curve in advance. Knowing which length MA to use exactly is

only clear in hindsight, but we don't need to know the exact shape in advance—we just need to know what the trends in direction are.

Because we normalize the predictions to price, the graph can become distorted over time. Let's look at the evolution of a shorter, more descriptive moving average, the 10 day, over time; this will show us what the MA described at the time and then what actually happened: price consistently followed predictions.

We begin on May 11. Price and prediction had been rising together in the first leg of Bitcoin's parabolic advance; here, prediction has made its first full, rounded peak since the beginning of the parabola, but price has continued to rise. Can we assume that price will follow?

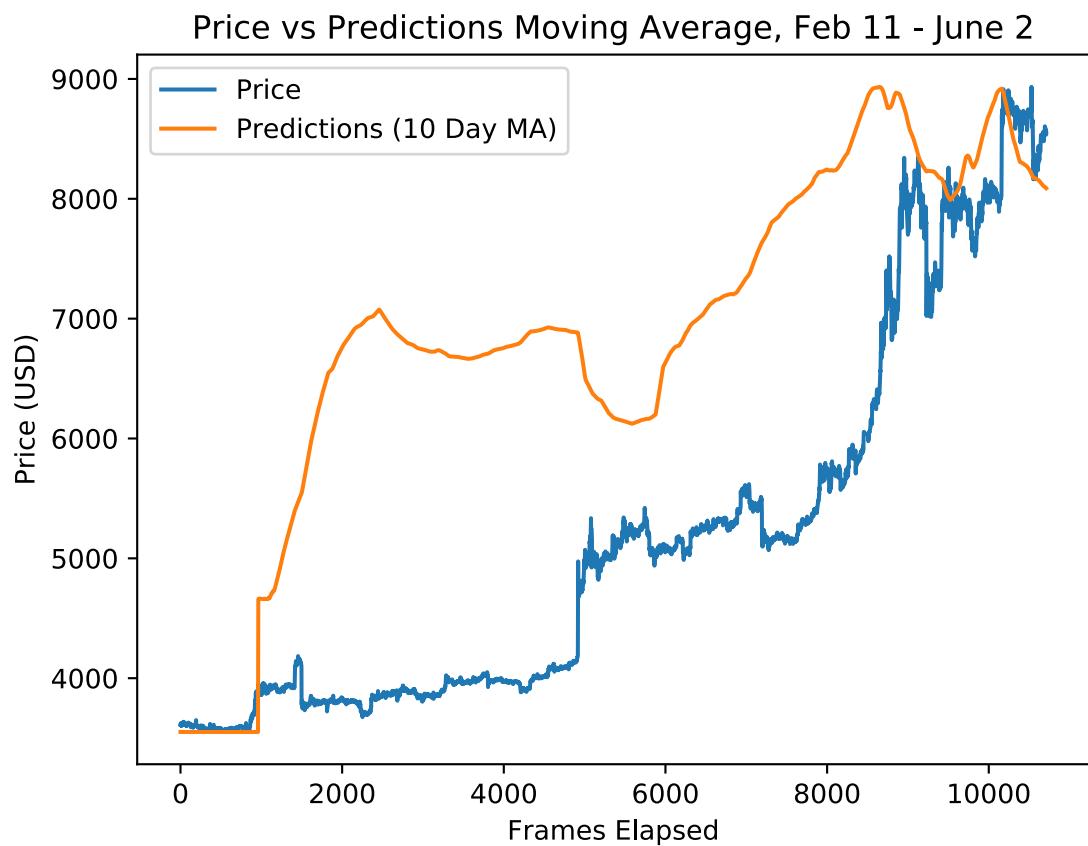Price vs Predictions Moving Average, Feb 11 - May 12
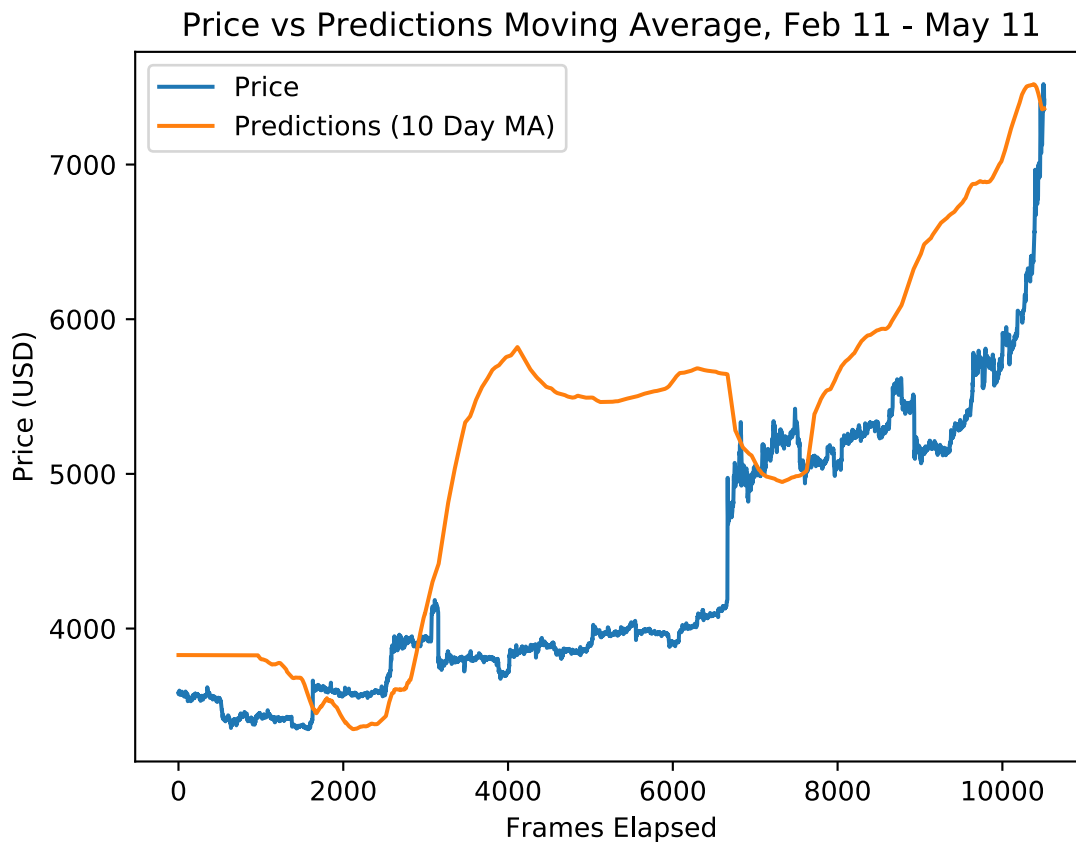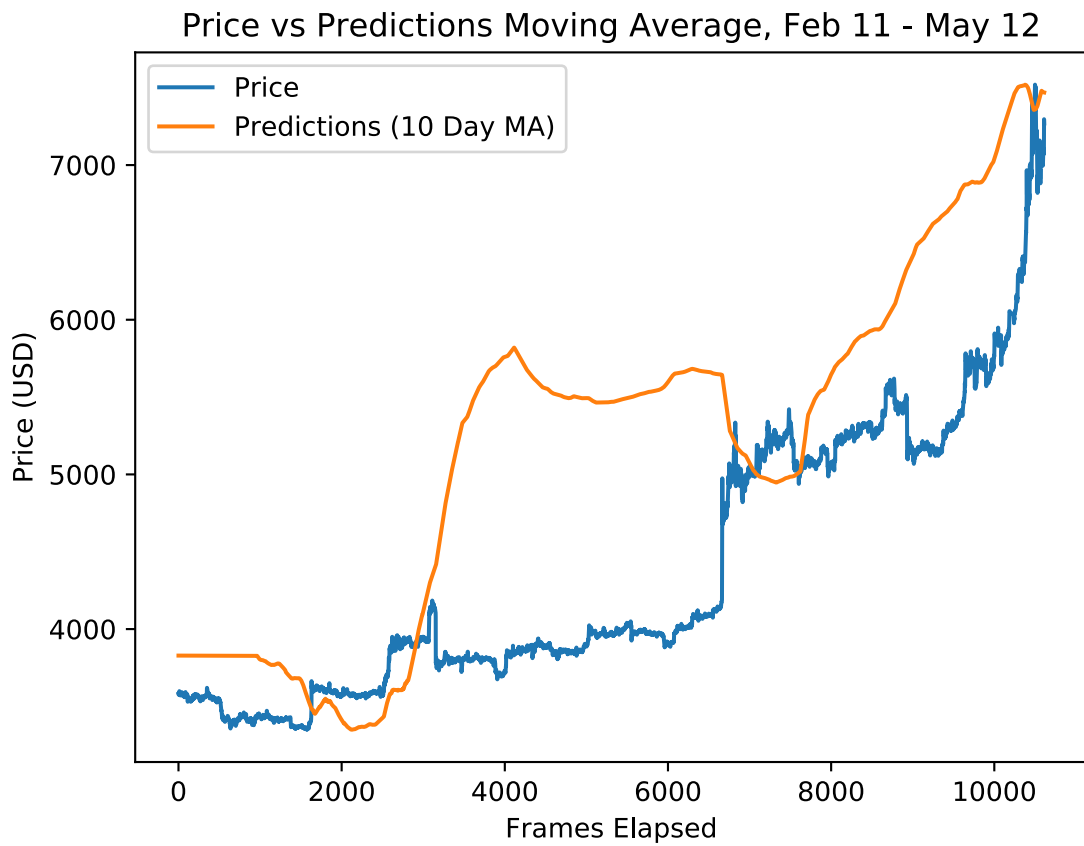
One day later (above), we see it did, briefly dropping below 7000. Simultaneously, predictions rise; price followed once again by rising above 8000.

By May 22, price was hovering, and predictions were rising…

Price vs Predictions Moving Average, Feb 11 - May 22

Price vs Predictions Moving Average, Feb 11 - June 2
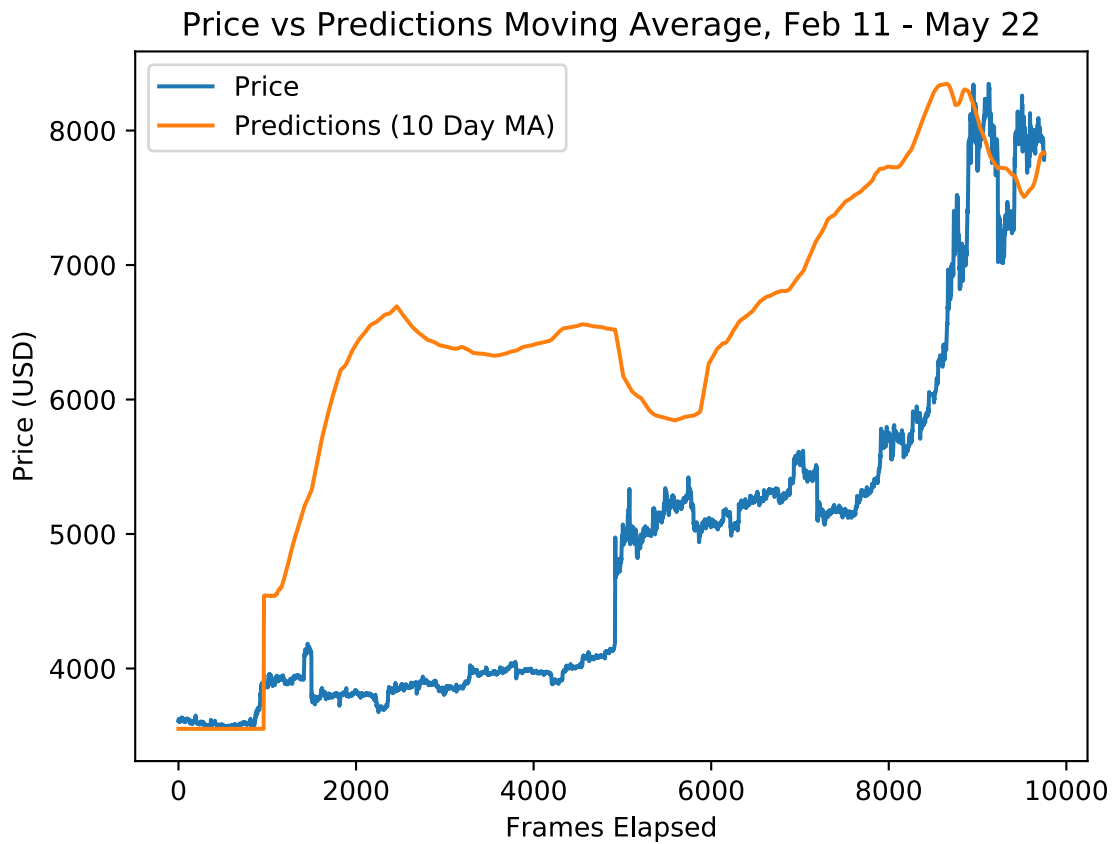
10 days later, price had shot up and predictions had begun trending down. Would price continue to follow predictions?

Yes; by June 12, price dropped to below 8000.

Price vs Predictions Moving Average, Feb 11 - June 12
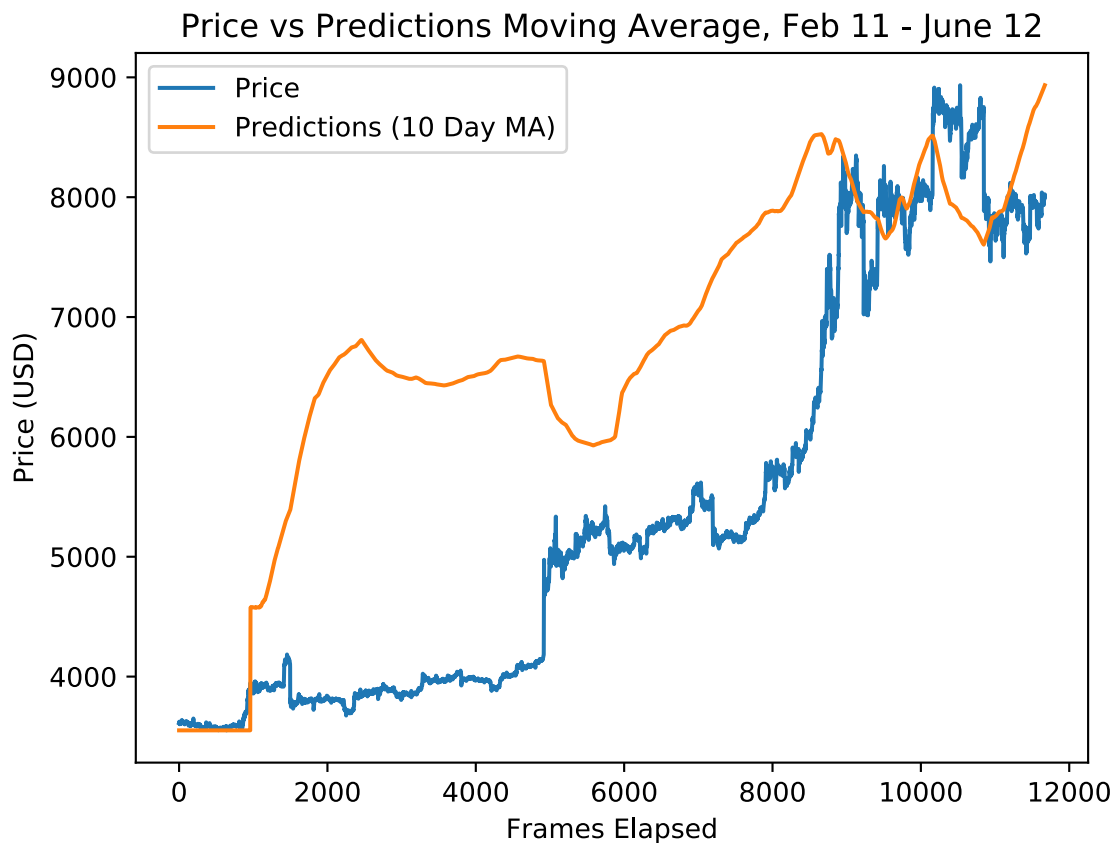
At that point, MA began making new highs, which, as we now know, were followed by new highs in prices a few days later—first by going above 9000 and eventually all the way to 13800.

To view a frame by frame progression of this sequence, use the "animate_ma" command in run_commands.py.

The contextualized, frame by frame description is meant to demonstrate how we, as a user of TradeBot at a given moment in time, might interpret its output. Instead of looking in hindsight at the shapes the moving averages paint, we focus on direction and magnitude—and see that price follows prediction.

Next, we look at the results of trading.

Trading Results—Strategy Variants

The pictures above are based on the full set of raw predictions. For trading strategies, we divided the raw data into different prediction sets.

From our cohort of regressions, we partition 4 different time frames: short term, medium term, long term, and "full set", which includes every regression. These time frames refer to the length of the period a given regression was trained on.

We also have 4 different prediction types: the raw prediction from the regression, a 'normal' prediction whereby the raw is adjusted to expected buy or sell thresholds, a 'raw optimized' prediction where we correct previous bias on a per regression basis for the raw prediction, and a 'normal optimized' prediction where we do the same for the normal. Note that these all come from the same value at any given moment, but the point at which they buy or sell may differ.

Lastly, we pool the predictions via either median or average. Hence, $4*4*2 = 32$ variants.

Note that you can simulate every strategy by using the "simulate_variants" command, and you can verify results with the time stamped files, which buy when prediction is above 1, and sell when prediction is below 0. The 1/0 buy/sell thresholds are determined by finding the previous most profitable path through a prediction set.

The following chart shows the performance of each strategy; values show a wallet which started at 100. For context, a wallet which bought and held Bitcoin since the beginning of our data set would have a value of 283.

Performance By Time Frame and Prediction Type

Metric	Raw	Normal	Raw Optimized	Normal Optimized	Average
Short Term Average	198.19	181.1	204.13	273.74	214.29
Short Term Median	200.12	130.94	204.13	273.74	202.23
Medium Term Average	202.79	181.55	260.27	225.46	217.52
Medium Term Median	167.32	192.78	279.66	154.42	198.54
Long Term Average	161.02	234.73	218.23	226.1	210.02
Long Term Median	192.66	162.28	209.02	175.92	184.97
Full Set Average	200.3	**286.94**	260.05	265.31	**253.15**
Full Set Median	216.17	217.85	242.51	258.14	**233.67**
Total Average	192.32	198.52	234.75	231.6	214.3

Observe a trend here in the bolded values of the right hand column: the significantly higher values for Full Set vs its counterparts and the total average. This confirms that predictions are not accurate only on certain time frames (e.g. short term or long term)—as we add more regressions to the set, the more accurate and valuable our trading becomes.

Overall, the results are positive: we have made a significant average profit in a short period of time. However, most variants have lagged Bitcoin's parabolic advance. The most successful to date, Normal Full Set Average (bolded in the "Normal" column), has outpaced Bitcoin's rapid advance. Let's examine its trade history to get a sense of how the system trades successfully:

Trade	Price	Date
Buy	$3780	2/24
Sell	$4666	4/2
Buy	$5400	4/25
Sell	$6966	5/11
Buy	$7237	5/12

Trade	Price	Date
Sell	$7915	5/13
Buy	$7447	5/16
Sell	$8817	5/27
Buy	$7885	6/3
Sell	$11143	6/22
Buy	$11660	6/27
Sell	$12443	7/8
Buy	$9980	7/14
Sell	$10600	7/18

There is good and bad in the performance here; it missed some moves and bought higher after a sell several times, but consistently sold higher than it bought, and did not get trapped in a high buy before the recent crash. It's a good example of how the TradeBot system can produce patience and long term holds, and also fast, responsive trading, while maintaining a high level of profitability.

Improving Trading Performance

The trading strategies above were launched in February; after several months of analysis, it's clear that despite their profitability, they are not the best way to leverage the accurate predictions. Several ways to improve:

—Eliminate the deadweight; switch to higher performing prediction sets only

—Instead of traversing prediction sets based on a best-path lookback, trade based on frequency of scores (a z score approach)

—Trade based on the expected value of any given prediction set: don't strictly eliminate lower performers, but weight trade size according to past success rate

—Trade based on rate of change in the full set moving average of adjusted predictions; this, based on our predictions, is most likely to take full advantage of our prediction knowledge

Improving Prediction Performance

To improve predictions, we need to continue to experiment with our model in a variety of ways, namely by testing different control variables which we have kept fixed to achieve our current results. This might mean:

—Adding a larger number of regressions

—Trying to predict smaller distances in the future
—Changing the way we sample data.

Generally, it means building out our capacity to experiment on a greater scale.

Conclusion

TradeBot's prediction engine offers robust predictions, best interpreted through a moving average, up to a week before the corresponding movement in price. The trading engine using these predictions, though profitable and in some cases matching Bitcoin's high performance, is not optimized to what it could be and will likely require changes to fully realize the value of the predictions.

Nevertheless, TradeBot has demonstrated its value live for 5 months in a verifiable way, and is a powerful tool for predicting and analyzing movements in Bitcoin prices. With further research and development, it could be more powerful still.