LEGAL GAMBLING: PREDICTING BITCOIN PRICES

ETHAN PALENSKE, ROWAN WILLIAMS

Abstract. In this paper we work to determine if we can predict future bitcoin prices if we have previous daily price values. We used two primary techniques: decision trees to determine if the price will be higher or lower, and autoregressive models to estimate the actual price. The goal was to use past data to correctly determine at least 80% of the time whether the next day's price had risen or fallen.

Unfortunately, both approaches performed far poorer than we hoped, with autoregression surprisingly doing even worse than guessing randomly. The tree-based models generally defaulted to always guessing that tomorrow's price would be higher. We examine a few reasons why this occurred and discuss ethical implications of financial modeling.

1. Research question and overview of the data

Our primary research question is if we can predict, at least 80% of the time, if the price of bitcoin will be up or down tomorrow using only past price data. Related to this, we want to determine which past prices are the greatest predictors of today's bitcoin price (1 day ago, 1 week ago, 1 year ago, etc.). We hypothesize that the best predictor of the price change tomorrow is the price change today, and that the relationship is negative due to the short-term reversal effect in finance. [J+25] The ability to predict future prices changes obviously gives a huge edge in investment markets.

There are very rarely any truly new ideas in finance, and most of these new ideas are bad ones. Price prediction has existed for as long as markets have existed, and everything we do in this paper has already been tried. One paper we examined used a wide variety of machine learning models to try to predict bitcoin prices, including random forests, XGBoost, autoregressive models, and more. [E+22] With the huge financial incentive that better price predictions yield, it is certain that every technique we implement here has already been tried.

The dataset consists of price of bitcoin recorded once each day for the past five years. At the time we downloaded the data, the dates ranged from April 25, 2020 to October 15, 2025. We do not know at what time of day the data was collected, and since there's only one data point per day the decisions are models will make will be based on data that is somewhat granular. However, this could may help our models avoid overfitting on more granular price oscillations. Also, we plan to have our models make

Date: November 25, 2025.

one trading decision per day and this dataset matches this frequency. Since bitcoin is the primary and most valuable cryptocurrency, its price is likely less chaotic than other cryptocurrencies, making this dataset potentially better for training than other options.

We expect our analysis to determine the best collection of past prices to look at in order to predict the price of bitcoin the next day, and how these days are related to future prices. For instance, we might find that if bitcoin was down yesterday, then its price would be up today, but if its price was up a week ago, it's likely to be up today.

2. Data Cleaning / Feature Engineering

For the most part, this dataset was pristine and easy to work with, requiring little modification. We began by downloading the bitcoin dataset as an excel spreadsheet. We exported the resulting dataset into a CSV file, and added column headers. As the results were stored as strings, we needed to cast the input index to dates and the input column to floats.

For some of the models, we needed to use the daily price change instead of the actual price data. As such, we created a new feature which was equal to the difference between consecutive bitcoin prices. This enabled us to train on the change in the price of bitcoin instead of training on the actual price, which is especially useful for the tree-based models we used. We also verified that no dates were missing in the pricing data, meaning we did not need to fill in any missing information. 1 Since our primary research question is determining if we can predict future prices using only past prices, we have no other features that we need to engineer.

```
# Reading in the dataframe and converting to dates and floats
btc_df = pd.read_csv("Data.csv", index_col=0, parse_dates=True)
btc_df["price"] = btc_df["price"].str.replace(",", "").astype(float)
btc_df["price_diff"] = btc_df["price"] - btc_df["price"].shift(1)
btc_ar = btc_df.values

# Verifying that no days are missing in the model
assert all(btc_df.index.to_series() == pd.date_range(start="2020-04-25", end="2025-10-15"))
```

Figure 1. Readying the dataframe for processing.

3. Data Visualization and Basic Analysis

In the graph of the dataset 2, we can see that the price of bitcoin has been generally increasing over the past few years. There are 2000 points in the dataset, and the price of bitcoin rose 1030 times and fell 969 times. These values are roughly equal, which means that the misclassification rate will be a good metric for our models. When bitcoin prices rose compared to the previous day they rose an average of $980, and when they fell they fell

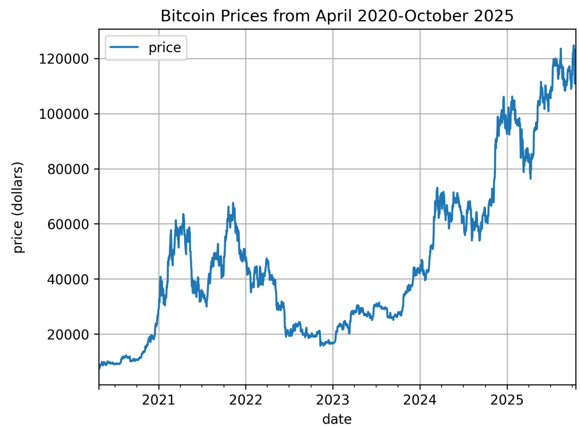

FIGURE 2. Daily bitcoin price data.

	Bitcoin Price	Daily Change in Price
Mean	47525.42	52.86
Std	29942.10	1498.79
Min	7495.39	-8496.29
Q1	23823.28	-469.65
Median	40508.71	22.70
Q3	63346.12	586.22
Max	124773.51	8256.27

TABLE 1. Summary statistics for bitcoin.

an average of $932. 1 This makes it clear that the price is rising on average, but is very volatile.

Because the data are fairly volatile, and prices increase or decrease at near-equal rates, this dataset is a good fit for working on this problem. An algorithm that guesses if the price will increase randomly will be our baseline, and we will try to get our algorithms to have a misclassification rate of 20% or lower.

4. Learning Algorithms and In-depth Analysis

Autoregression. The first type of model that we used is an autoregressive model. This is a type of linear regression model where the variables are past values of the timeseries data. In essence, autoregressive models are like a generalized Fibonacci sequence; the next term is estimated by a sum of scaled previous terms. Autoregression picks the optimal coefficients for this linear model. [fG25] The term “factor” is used to refer to a particular variable that looks back a certain distance.

For instance, one model could be of the form $P_t = 1.1P_{t-1} - 0.2P_{t-2}$. This model predicts that the next price is a scaled sum of the previous two prices. P_{t-1} and P_{t-2} are factors, while 1.1 and -0.2 are trained coefficients.

We tried training these models to predict both kinds of features in the dataset; the future price of bitcoin and the future return of bitcoin, or the price change between two days. We used a train-test split to verify that the model was tested on data that it had no trained on. Modeling tomorrow's price was somewhat successful. 3

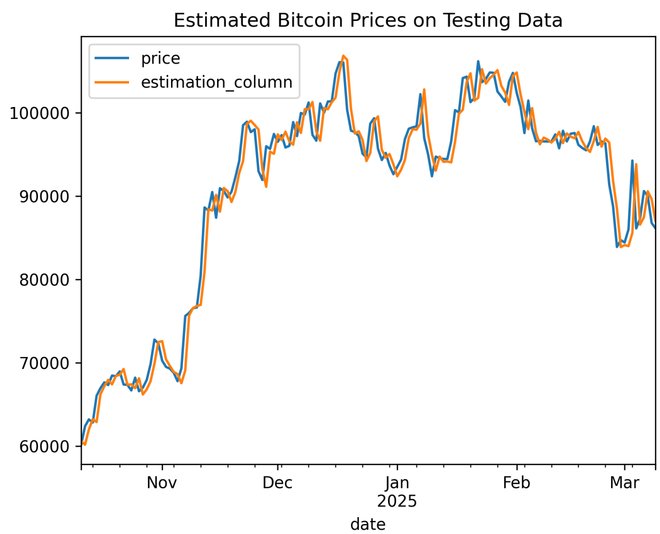

FIGURE 3. Comparing true bitcoin prices with prices estimated by an autoregressive model. This model looked at the 30 prior days and trained coefficients to scale each one by. The estimated line appears to be lagging since the best predictor of tomorrow's price of bitcoin is today's price.

While the results look graphically impressive, the model was ultimately not useful for making trading decisions, due to the lag in the estimated value. Upon running autoregression on the daily change of the price of bitcoin, the results were disappointing. 4

In fact, we checked to determine how often the sign of the result matched the sign of the true change in price (i.e., if the model accurately predicted if we should buy or sell), and the model gave the correct answer only 48% of the time. Experimenting with different intervals of time and different amounts of variables in the linear regression brought this percentage up to 52%, which is 0.5% better than a model that always guesses that bitcoin will be more valuable tomorrow.

This exceedingly poor performance is likely due to the fact that autoregression models are designed to match specific future values, not determine a binary classification like the one we are examining. Similarly, due to the

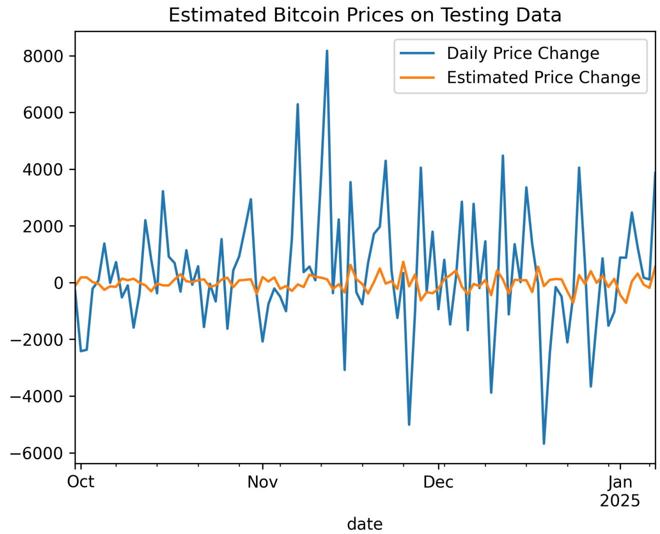

FIGURE 4. Comparing the daily change in bitcoin prices to the estimated daily change. It is clear that the estimated values are nowhere near the true values, and in fact, they share the same sign less than half of the time. The rate of sign-matching improved only 4% after experimenting with different choices of factors to include in the model.

high volatility of bitcoin prices from day to day, the graph of returns is very messy and sharp, making it difficult to determine a linear relationship between past returns and tomorrow's return. Thus, it makes sense that this classifier performed poorly.

In light of these poor results, this model was not able to answer any of our research questions, except to clarify that autoregressive models *can't* accurately predict future bitcoin prices on their own.

Trees and Forest Classifiers. The equity curves illustrate how each classification based trading strategy performs against the buy-and-hold baseline over the evaluation period of 2025. The baseline (in red) ultimately achieves one of the highest terminal equity, unconvincingly surpassed by only two models, and shows a relatively smooth upward drift, which is consistent with Bitcoin's general tendency to rise in value over time. The fact that buy-and-hold dominates most of the classifier-driven strategies indicates that the models were not reliably identifying periods of positive return better than simply staying continuously invested.

Among the decision-tree models, T-C2 tracks close to the baseline early on but diverge downward, and fails to recover. The random-forest models show somewhat more resilient performance: generally following more closely

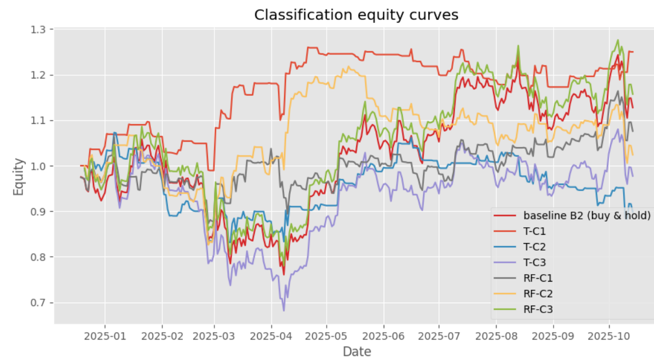

FIGURE 5. Equity classification performance of tree and forest models. "T and RF" indicate Tree and RF respectively, "C" merely distinguishes different models of the same type with an index number

to the baseline strategy with slight deviation in the beginning that carries over to the end

Overall, the figure reinforces two points. First, none of the classification models consistently outperformed the baseline, so their buy/no-buy signals were not capturing any structure. Second, even the best models provided only marginal protection during downturns and did not meaningfully enhance gains during a "bull market". As a result, the classification strategy as implemented here did not produce improvements over alternative predictive frameworks.

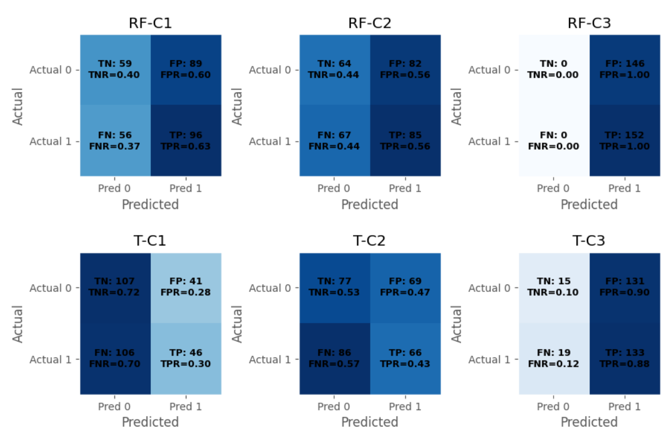

FIGURE 6. Confusion matrix for all classifiers.

In reviewing the confusion matrices across all classifier variants, we found that nearly every model was overwhelmingly biased toward predicting BUY, in some cases doing so 100% of the time. This pattern indicates that the classifiers were unable to learn a meaningful boundary between BUY and NOT-BUY classes from the features provided. Instead, they defaulted toward the majority outcome: BUY (the market trends upwards).

This essentially means that a model can outperform the market by getting one or two guesses right in the beginning and then holding (which is exactly what the baseline strategy is) the rest of the testing period. Any model that does not do this generally performs worse than the market, so it does not get selected for being good enough to keep results for.

Put more directly, single decision trees overfit and performed poorly, and random forests mitigated some of that instability further by acting too conservatively; both families still produced heavily BUY-skewed predictions, meaning that the available features did not offer clear separation between the two classes. As a result, the raw classification metrics—especially those tied to a default decision threshold of 0.5—do not provide an accurate picture of true predictive capability.

5. Ethical Implications and Conclusions

Is it ethical to sell a security to someone if you are sure its value is going to go down? Is it moral to make money off the uninformed decisions of others? These questions are at the core of finance, and while they don't have simple or short solutions, but they are important to consider.

The models that we studied here are excessively simplistic. The best predictive models must incorporate more information than just past bitcoin prices. Current shocks to the markets, legislation that impacts cryptocurrency trading, and news that impacts public opinion will have a huge impact on bitcoin prices, but can't be predicted just from past bitcoin data.

As George E. P. Box once said, "All models are wrong, but some are useful." Anyone who uses financial models needs to be informed of their frequent fallibility and the information, or lack thereof, that the model incorporates. Fortunately, with so many people invested in making the optimal financial decisions, individual, hobby models rarely cause issues, and few people pay attention to any one model for that model to cause any destructive feedback. Similarly, since the price data are public, there are no privacy concerns related to using it.

We end with a quote from Matt Levine, a financial columnist and advisor for Bloomberg, who brings up an issue with financial modeling in general. "If people devote a lot of skill, intelligence and knowledge to financial markets, then those skilled, intelligent, knowledgeable people aren't doing something else.... In any case, though, many smart people who are good at solving problems are solving problems like "how can we quote tighter bid/ask spreads on interest-rate futures" rather than problems like "how can we cure cancer," because the money is better (and the feedback more immediate) with the interest-rate futures." [Lev25]

References

[E+22] Sahar Erfanian et al. Predicting bitcoin (btc) price in the context of economic theories: A machine learning approach, 2022.

[fG25] Geeks for Geeks. Autoregressive (ar) model for time series forecasting, 2025.

[J+25] Narasimhan Jegadeesh et al. Short-term reversals and longer-term momentum around the world: Theory and evidence, 2025.

[Lev25] Matt Levine. Money stuff: Quantum bond trading, 2025.