Bitcoin Price Prediction Using Penalized Regression

Project 1

Trey Schulman (treys3)

09/29/2025

Table of Contents

1. Introduction

The purpose of this project is to build and evaluate predictive models for Bitcoin's market price a week in advance using blockchain and market-related features. Predicting Bitcoin's price was an interesting challenge due to high multicollinearity among blockchain metrics, heavy-tailed distributions, and the volatility of cryptocurrency. To address these challenges, the project methodology combined exploratory data exploration, data cleaning, feature engineering, and model training with penalized regression approaches. Ridge, LASSO, and Elastic Net methods were chosen to hopefully mitigate multicollinearity and improve predictive performance, with model selection guided primarily by validation RMSE.

2. Data Configuration

The dataset originates from Kaggle's cryptocurrency price history and contains daily Bitcoin-level metrics spanning 2010 to 2018. Variables include market indicators (e.g., market price, trade volume), network activity metrics (e.g., number of transactions, unique addresses, hash rate), miner revenue measures, and transactional statistics.

2.1 Data Cleaning

Early periods of the dataset recorded Bitcoin's market price as zero, reflecting a time when trading was likely not yet established. These zero-price days were removed to avoid introducing bias into the model. The variable btc_trade_volume contained 21 missing values scattered throughout the series. These were imputed using forward fill (ffill), which maintains continuity without introducing abrupt changes. This choice is justified because trading volume exhibits persistence over time. Highly redundant or deterministic variables were also removed. For example, btc_market_cap was excluded since it is a product of supply and price, and btc_miners_revenue was dropped due to strong collinearity with fees and difficulty. This multicollinearity is demonstrated respectively through Figure 1 and Table 1 showing high correlation and variance inflation factors. Other cumulative counters

such as btc_n_transactions_total were excluded because they acted as proxies for time.

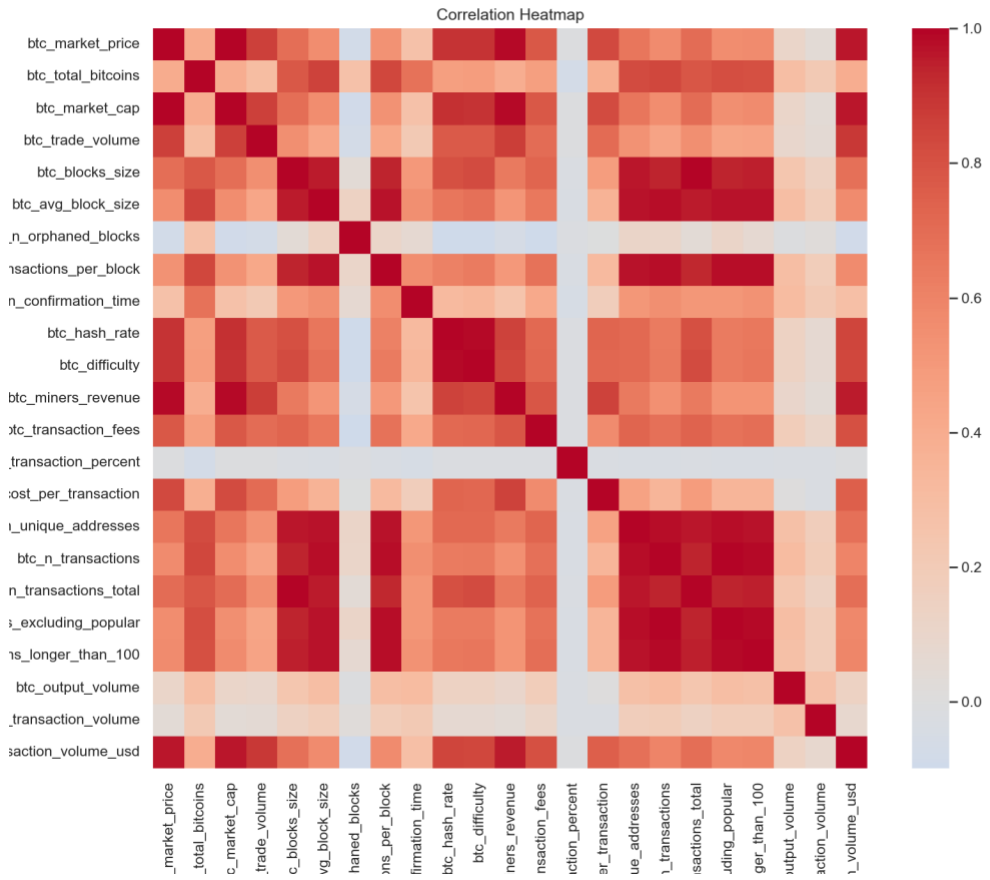

Figure 1: Correlation Heatmap

feature	VIF
btc_blocks_size	9,049.2
btc_n_transactions_total	9,036.1
btc_market_cap	259.8
btc_miners_revenue	249.2
btc_difficulty	231.0
btc_n_transactions	216.3
btc_hash_rate	159.8
btc_n_transactions_excluding_popular	131.7
btc_n_transactions_excluding_chains_longer_than_100	103.2

Table 1: Variance Inflation Factor

2.2 Transformations

Several variables exhibited heavy right skew and large spikes. To stabilize variance and reduce the influence of extreme trading days, log transformations were applied to btc_trade_volume, btc_transaction_fees, btc_output_volume, and btc_estimated_transaction_volume. These transformations were implemented to improve linearity and make penalized regression more effective.

3. Feature Engineering

Feature engineering focused on constructing predictors that are informative and leakage-safe. Autoregressive features included lags of market price at 1, 7, 14, and 30 days, as well as rolling averages and rolling standard deviations to capture local trends and volatility. Calendar dummies for day-of-week and month were also created to model potential cyclical effects in the data. The target variable was defined as target_t7, representing the market price seven days ahead. The final feature engineered dataset ended with 2,708 records and 45 fields, which are detailed in the Appendix 1 item.

4. Model Selection and Prediction

The dataset was split chronologically into 70 percent training, 15 percent validation, and 15 percent testing sets. Within the training data, a five-fold expanding-window TimeSeriesSplit was used to tune hyperparameters. This train/validation/test approach prevented leakage across time and mirrors real forecasting conditions while protecting the integrity of the test set.

The models considered included ordinary least squares (OLS), Ridge regression, LASSO regression, and Elastic Net. OLS was used only for diagnostic purposes, reporting Adj R^2, AIC, and BIC, but was not considered reliable due to multicollinearity. Penalized regressions were the focus, since they shrink or eliminate correlated predictors.

The primary selection metric was validation RMSE, but MAE and R^2 were also reported as secondary measures. OLS metrics were reported separately for discussion. RMSE was chosen as the primary selection metric because it punishes large deviations and allows direct comparison across different methods on a single meaningful scale. LASSO emerged as the strongest model, achieving the lowest validation RMSE. On the test set, the LASSO model achieved an RMSE of approximately 1286, an MAE of 714, and an R^2 of 0.92.

model	split	rmse	mae	r2
lasso	val_lasso	56.34	42.71	0.86
ridge	val_ridge	108.81	87.27	0.48
elasticnet	val_elasticnet	280.51	254.78	-2.48

Table 2: Validation Metrics

feature	coef
btc_market_price	246.43
btc_n_transactions	3.00
btc_market_price_std7	0.25
btc_trade_volume_log	0.06

Table 3: Selected Lasso Features & Coefficients

5. Discussion

These results demonstrate the value of penalization in this context. OLS produced high in-sample fit, but its coefficients were unstable due to multicollinearity and overfitting. Ridge and Elastic Net provided more stability, but LASSO combined strong predictive accuracy with parsimony by selecting a small set of four key predictors. The only features retained by LASSO included lagged market price, seven-day rolling volatility, log trade volume, and number of transactions. Residual plots revealed that errors widened during periods of extreme volatility, reflecting the limitations of linear models.

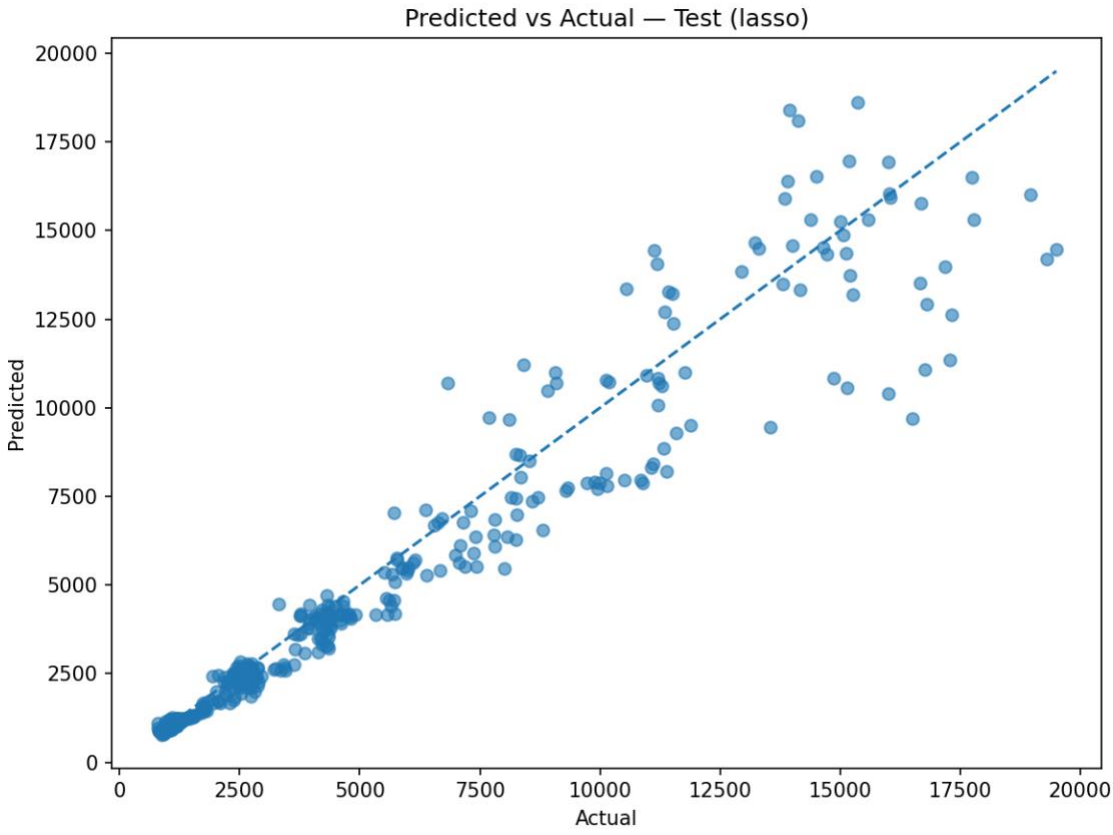

Figure 2: Lasso Predicted vs. Actual

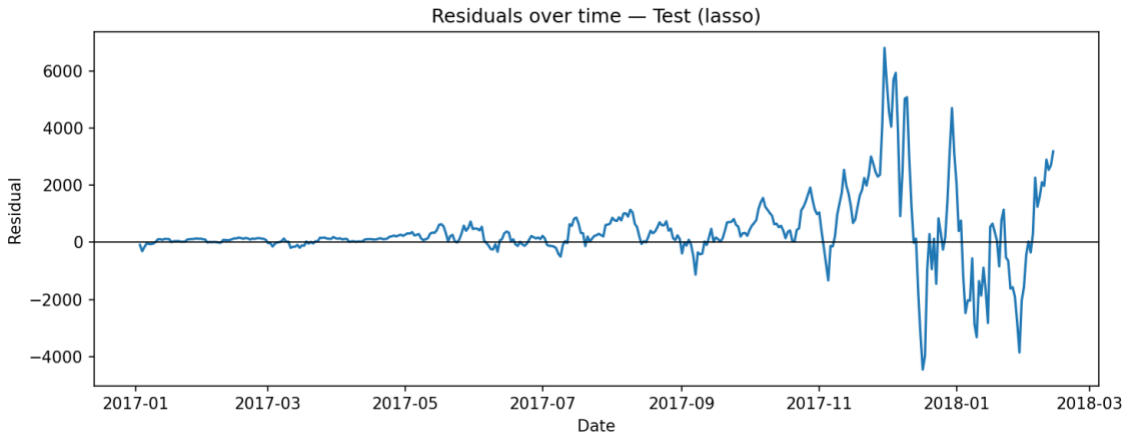

Figure 3: Lasso Residuals over Time

For practitioners or traders, these findings suggest that near-term price forecasts benefit from including both price persistence and on-chain activity measures. However, the models are less reliable in turbulent market conditions.

6. Conclusion

The best-performing model was LASSO regression, which balanced accuracy and interpretability. Its results highlight the importance of combining autoregressive structure with blockchain activity signals. Nevertheless, prediction errors remain large relative to the scale of Bitcoin prices. Future work could investigate nonlinear models such as tree ensembles or neural networks, as well as rolling re-fit schemes to adapt to changing market regimes.

Appendix

```
Index(['Date', 'btc_market_price', 'btc_total_bitcoins', 'btc_trade_volume',
       'btc_avg_block_size', 'btc_median_confirmation_time', 'btc_hash_rate',
       'btc_transaction_fees', 'btc_cost_per_transaction',
       'btc_n_unique_addresses', 'btc_n_transactions', 'btc_output_volume',
       'btc_estimated_transaction_volume', 'target_t7',
       'btc_market_price_lag1', 'btc_market_price_lag7',
       'btc_market_price_lag14', 'btc_market_price_lag30',
       'btc_market_price_ma7', 'btc_market_price_ma14',
       'btc_market_price_ma30', 'btc_market_price_std7',
       'btc_market_price_std14', 'btc_market_price_std30', 'dow_1', 'dow_2',
       'dow_3', 'dow_4', 'dow_5', 'dow_6', 'month_2', 'month_3', 'month_4',
       'month_5', 'month_6', 'month_7', 'month_8', 'month_9', 'month_10',
       'month_11', 'month_12', 'btc_trade_volume_log',
       'btc_transaction_fees_log', 'btc_output_volume_log',
       'btc_estimated_transaction_volume_log'],
      dtype='object')
```

Appendix 1: FE Fields