report

December 17, 2025

0.1 Predicting 30 Minute Bitcoin Volatility

0.1.1 Motivation

Financial time series provide an abundance of data, within which one can find a number of interesting and difficult modeling problems. For instance, if one can predict returns of an equity over some time horizon, it may be possible to trade on that signal. Another well-known characteristic of these datasets is realized volatility, which we define as

$$\sqrt{\frac{1}{T}\sum_{t=1}^{T} r_t^2} \times \sqrt{24 * 60 * 365}$$

where r_t is minutely returns, and T is some time horizon (which we took to be 30 minutes). We annualize volatility to make it comparable over different time horizons. If one can predict volatility well, it allows for better options pricing, which again may lead to a profitable trade. There is a rich literature on volatility modeling, but most models - such as GARCH - consider much longer time scales. Predicting short-term volatility requires more data and more complex models, so is an appropriate choice for a data science project. Furthermore, volatility has a much higher signal to noise ratio than returns, so a high performance model is feasible without proprietary data.

We chose to model volatility for Bitcoin, for three reasons. Firstly, high-quality granular data is easily accessible through the Binance API. Stocks may trade on multiple exchanges and often have data quality issues. Secondly, Bitcoin has no "fundamentals", unlike equities. This simplifies modeling, since there are no company-specific variables to consider. How Lastly, Bitcoin options trade on prediction markets, so predicting forward looking realized volatility for time scales less than 1 hour can provide edge for a trading strategy on those contracts.

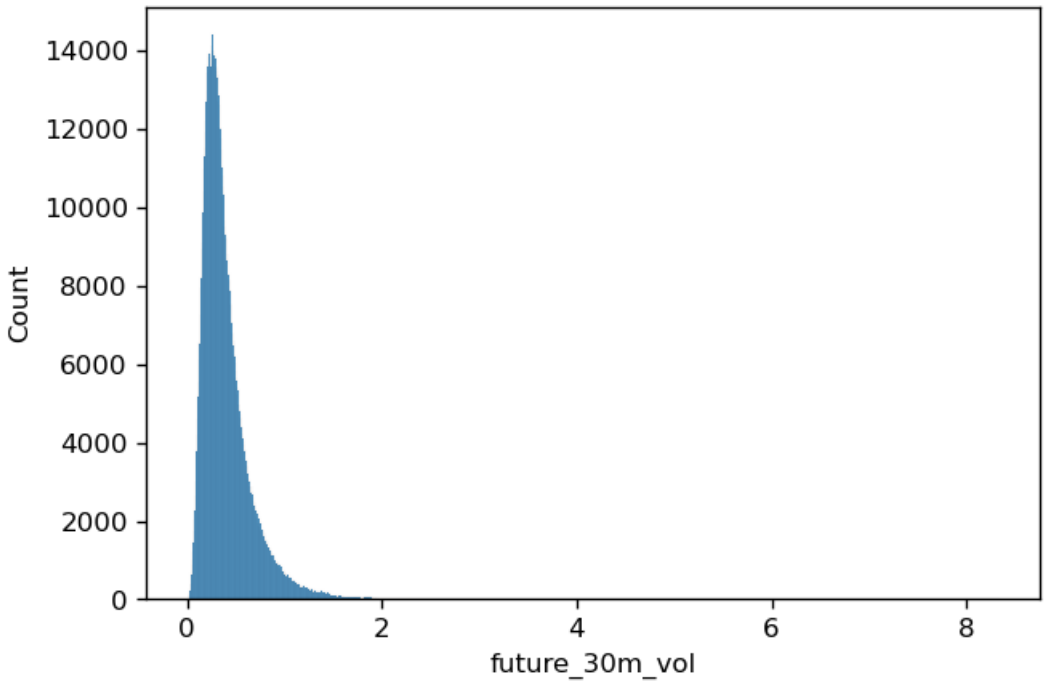
The responder and most of the features used in this project are calculated directly from the raw Binance data, which is again an interesting challenge; compressing data over multiple time scales into usable features requires intuition for the problem at hand. Also, to ensure the model is usable in practice, one must be careful to avoid data leakage from past to future. This problem arises in feature engineering, cross-validation, and fitting transformers. This consideration is implicit in all of the modeling choices that were made throughout the project.

0.1.2 Explanatory Data Analysis

Our exploratory data analysis (EDA) was slightly unconventional in the sense that we did not have pre-defined features and responders. We first had to create the volatility responder, explore its relationship with potential features, then decide on a finalized dataset. As mentioned before, the data was very clean, so our primary focus was on manipulation/feature creation as opposed to handling missing values or other common problems.

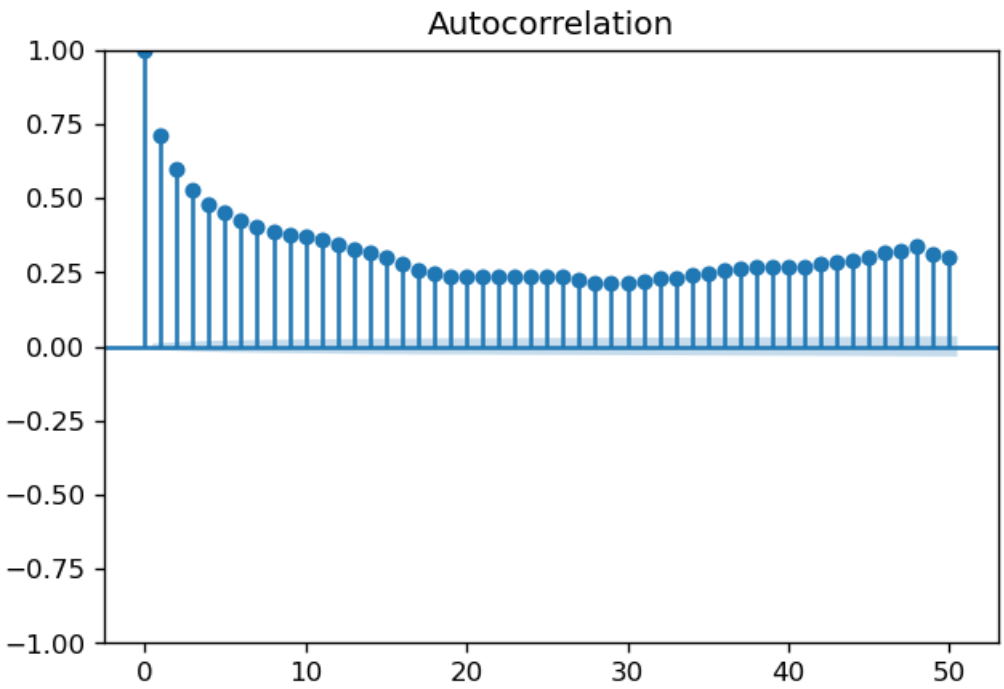
In this section, we motivate our modeling choices by presenting key findings from the EDA notebook. First, we discuss the raw data itself. We pulled roughly 2 years of minutely price data from the Binance "klines" API. The data of interest was open price for each minute, the number of trades that occurred, and the amount of Bitcoin in US dollars that traded over the minute (quote asset volume).

Now, we look at the distribution of 30 minute volatility to begin our *explanatory* data analysis.



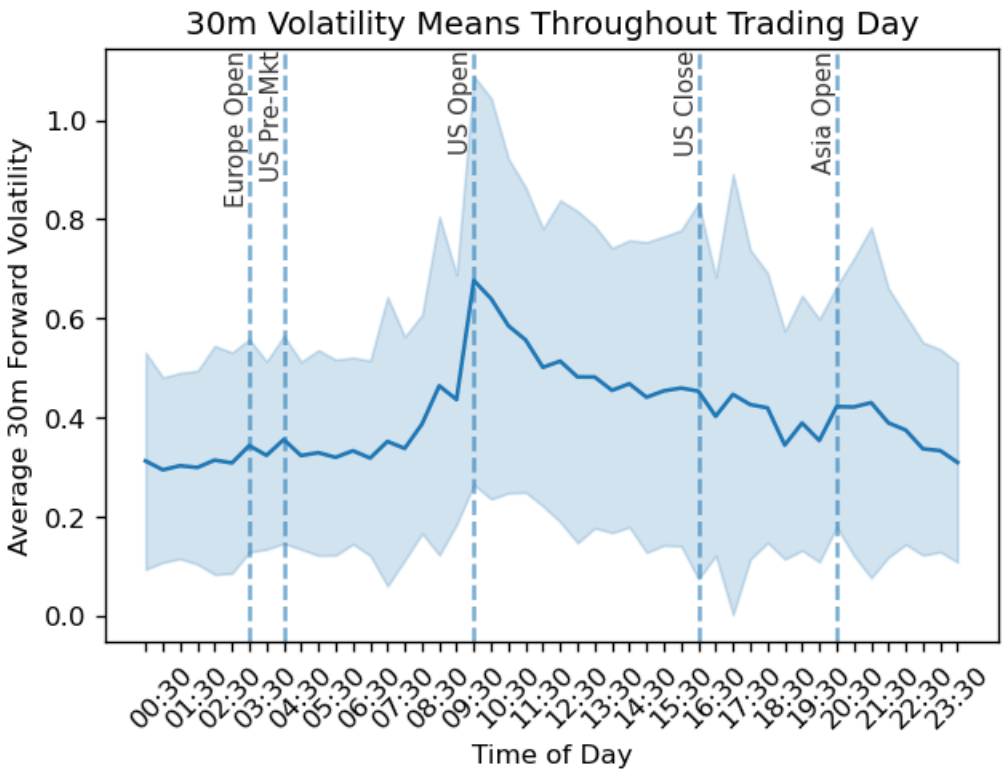
The responder has a smooth distribution with a heavy right tail. This is logical, since returns are roughly mean zero with heavy tails, and volatility is a function of squared returns. This justifies our use of a gamma glm, and gamma deviance for both the glm and lgbm model. These choices enforce the positivity of the responder, and allows variance to increase with the mean. However, we don't expect the glm to be a perfect fit, since the relationship between our features and responders is heavily constrained. The distributions of other features (not graphed here for brevity) show similar patterns. The mean-zero features have long tails on both sides, and the strictly positive features have long right tails. This motivates our decision to try models with both clipped and unclipped features in the model evaluation notebook.

Next, we look at the autoregressive properties of volatility. Is current volatility a good predictor of future volatility? To assess this, we used an autocorrelation plot, which gives the correlation between a time series and multiple lagged versions of itself. We downsampled to 30 minute windows, so each observation is independent.

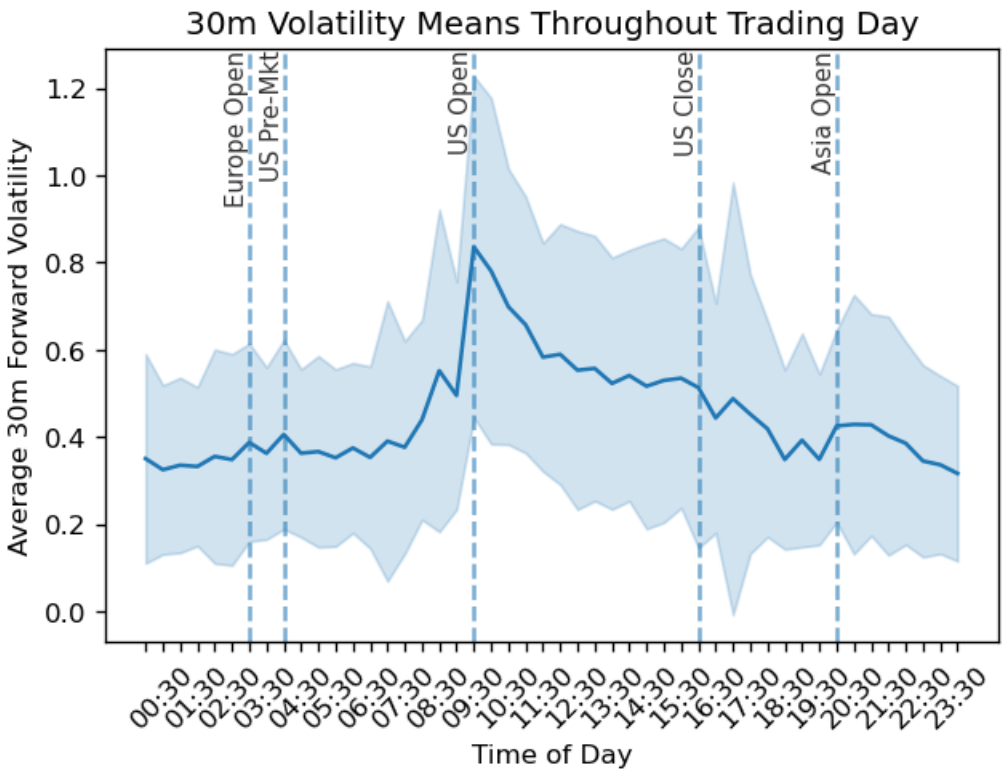


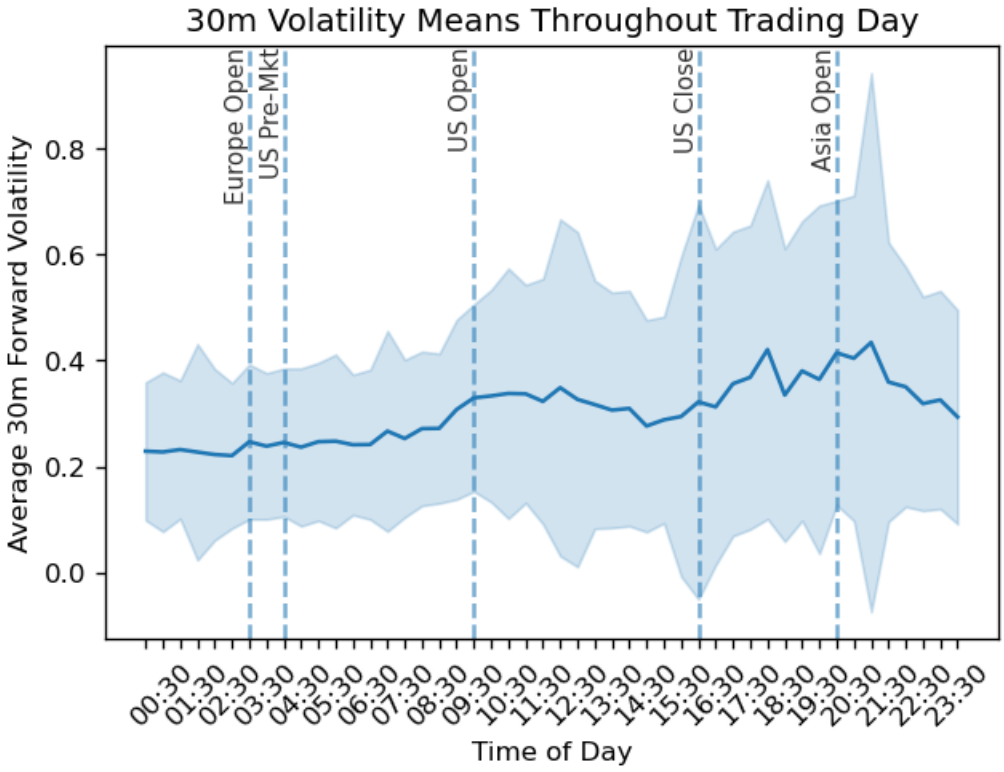
There are two interesting things in the graph. First, the correlation between the previous period's volatility and the next is close to 0.75. This steadily decays to near 0.25 12 hours later. Then, after 24 hours, we see another increase in correlation! This informed our hypothesis about daily time trends: there may be periodicity in the time series, with patterns recurring each day.

Therefore, the next thing we looked at the relationship between volatility and time of day. It is well known that Bitcoin returns are highly correlated with equities returns. Each day, trading on the NYSE opens at 9:30 AM ET. This marks a period known as "price discovery", where overnight information is incorporated into prices at the open. This is a high-volatility time for equities, so we expect to see increased volatility in Bitcoin as well. The following graph confirmed our hypothesis:



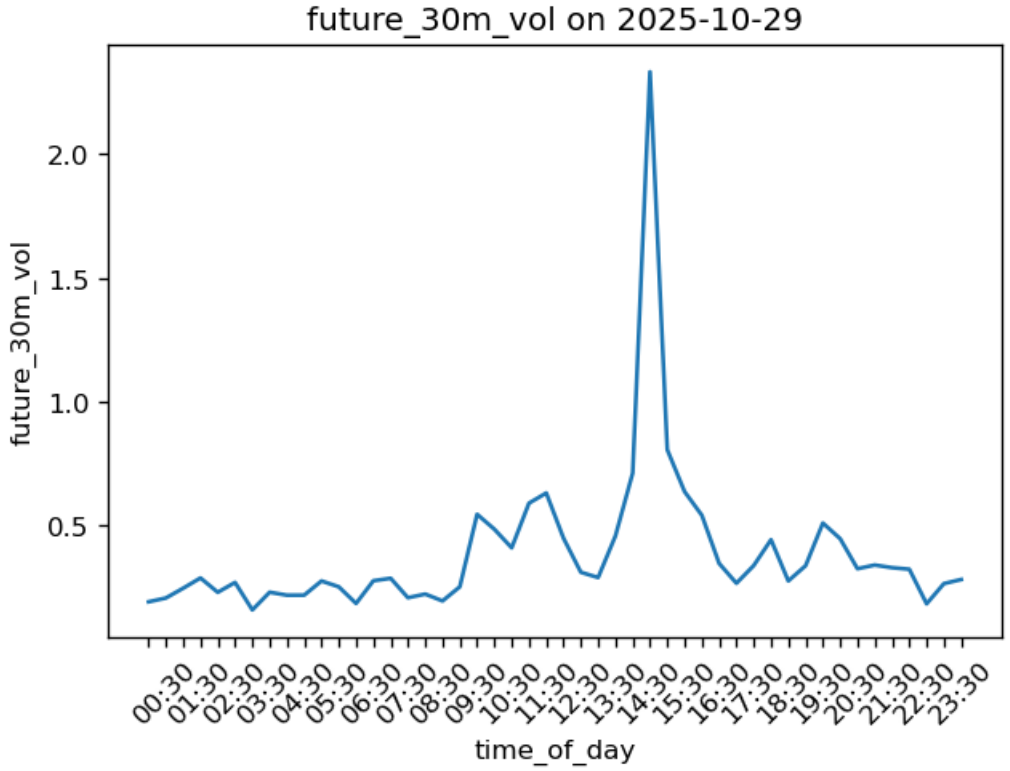
We see pronounced spikes at the US open, with smaller increases corresponding with open times of other markets. We expect this relationship to hold on trading days, but not on weekends or holidays. Again, this difference is shown in the following graphs.


30m Volatility Means Throughout Trading Day
Europe Open
US Pre-Mkt
US Open
US Close
Asia Open
Average 30m Forward Volatility
0.0
0.2
0.4
0.6
0.8
1.0
1.2
00:30
01:30
02:30
03:30
04:30
05:30
06:30
07:30
08:30
09:30
10:30
11:30
12:30
13:30
14:30
15:30
16:30
17:30
18:30
19:30
20:30
21:30
22:30
23:30
Time of Day

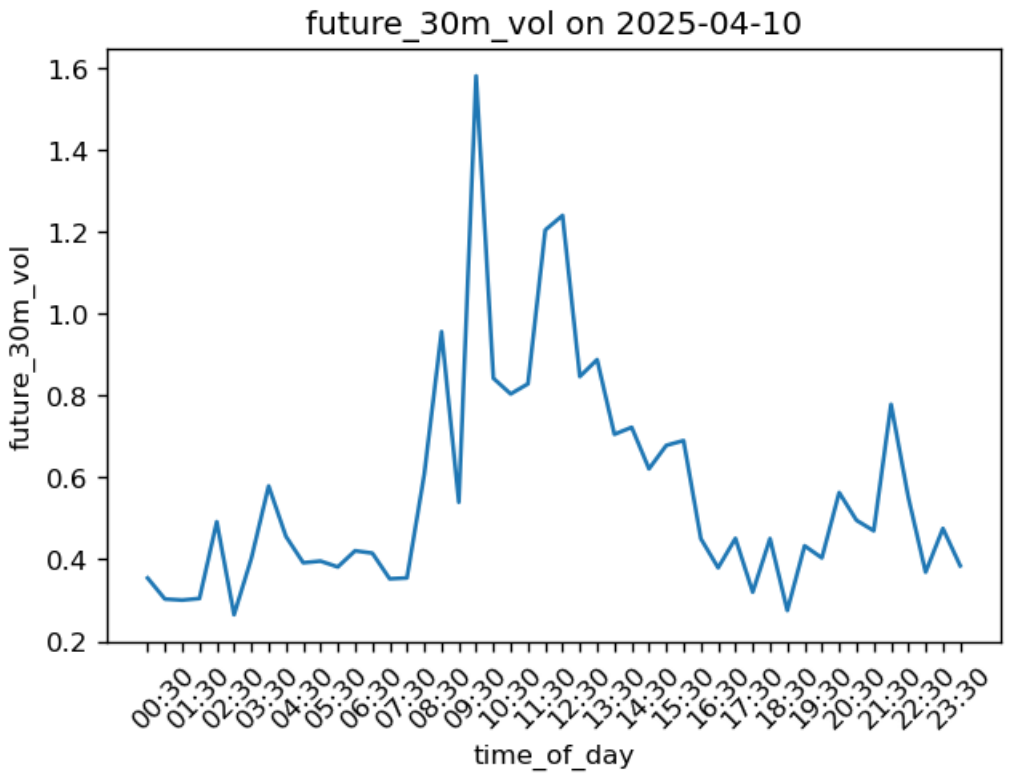

30m Volatility Means Throughout Trading Day
Europe Open
US Pre-Mkt
US Open
US Close
Asia Open
Average 30m Forward Volatility
0.0
0.2
0.4
0.6
0.8
00:30
01:30
02:30
03:30
04:30
05:30
06:30
07:30
08:30
09:30
10:30
11:30
12:30
13:30
14:30
15:30
16:30
17:30
18:30
19:30
20:30
21:30
22:30
23:30
Time of Day

It is clear that the relationship breaks down on non-trading days, where we see volatility increase throughout the day without a defined spike at market open. This motivates our use of time of day, and market trading calendars, as categorical features for our model.

We guessed that there may also be a relationship between volatility and macro events, such as rate cuts or the release of jobs data. To investigate this, we chose two dates to spot check. The first was October 29, 2025. At 2 PM ET, the fed announced a decision on rates.



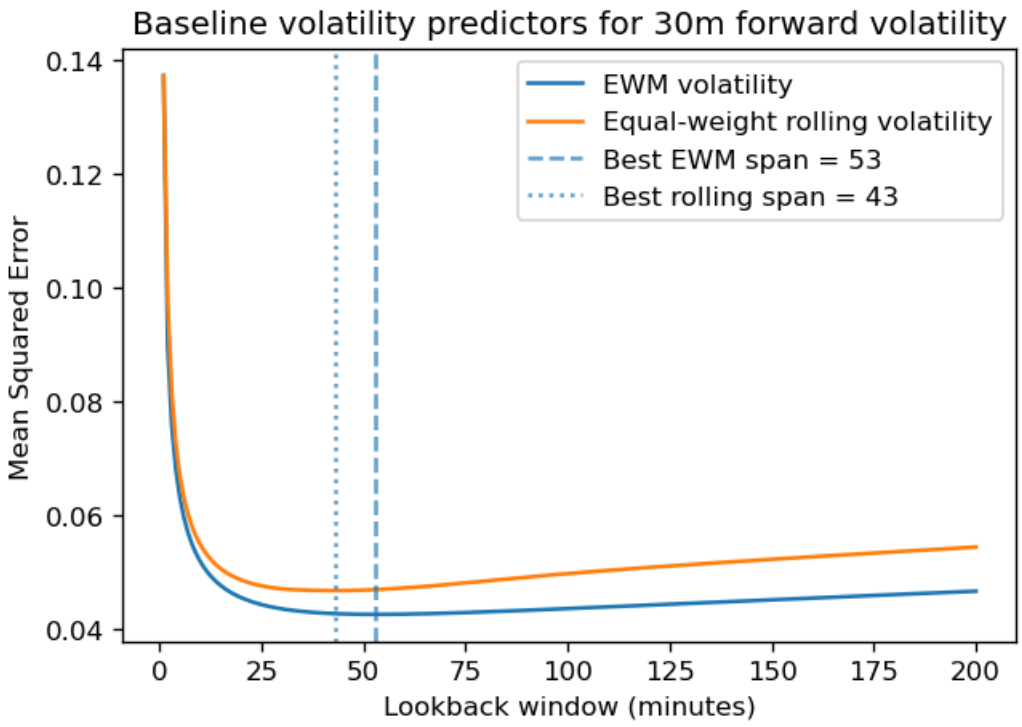
We observe a large spike in volatility on an otherwise uneventful day. Next, we looked at April 10th, when the consumer price index (CPI) data was released at 8:30 ET.



There is clearly an increase in volatility 1 hour before the open, but nowhere near as large as the fed announcement. Thus, we decided to include fed announcements as well as unemployment and CPI data releases as categorical features.

Lastly, we motivate our usage of an exponentially weighted moving average (ewm) of squared minutely returns as a baseline predictor. One might expect that more recent data will have a larger impact on future information than "stale" data. Thus, an ewm might be a better predictor of future volatility than a simple historical rolling mean. We have already seen that past volatility is highly correlated with volatility in the next period. But what backwards-looking window makes the best prediction?

The graph below plots the mse of predictions based on a single rolling average on the y-axis, with the length of the window on the x-axis. The "span" of the ewm can be thought of as the effective length of the window.



Unsurprisingly, the ewm volatility dominates the rolling average over every time period. The MSE curve demonstrates the bias-variance tradeoff when considering past windows. Short-term returns are fresh information, but very noisy. Long-term averages are more stable, but may not reflect recent market moves. Thus, we chose a 50-minute span ewm of past squared returns as a simple feature for our models, as well as a baseline predictor to measure our predictions against.

Data Cleaning Steps Our data had no missing or incorrect values before feature construction, so we focus on transformations and data engineering steps taken before modeling.

- **Dates and times**. We stored all timestamps in Eastern Time as strings in lexigraphical order, to ensure we could maintain order of the dataset. Oftentimes, parsing strings is more straightforward than datetimes from different libraries. We also extracted| the date, hour, and minute for each row, as well as the "time_of_day", to be used as a categorical feature.
- **Addition of categorical features**. To add data releases to the dataset, we merged them with the Bitcoin data on timestamp. We labeled rows without events as "NONE". We similarly imported trading calendars for different markets, aligned the format of the timestamps, and merged.
- **Lagged features**. Many of the features we constructed required shifting the data, which created nans at the beginning or the end of the dataset. These were dropped before modeling.
- **Clipping**. Due to the fat tails of the features and responder, we saved two clean datasets: one with Winsorized features, and one with raw features. The values for clipping were set on an early portion of the dataset to prevent data leakage.

0.1.3 Feature Selection and Engineering

From our EDA, we decided to include five categories of features:

1. Historical Volatility
2. Historical Returns
3. Metadata
4. Calendar and Time
5. Events

Historical Volatility Recent volatility is clearly a good indicator of future volatility - this finding is encoded in our 50 minute ewm of squared returns. We also hypothesized that there may be long-run averages, over the span of weeks or months, that may be helpful for a model to detect mean reverting patterns. Thus, we included one-day, one-week, and one-month previous realized volatility estimates.

We also expected that recent changes in volatility may inform the trajectory of volatility in the near future, so we included percent change of the ewm volatility relative to a recent baseline.

Historical Returns Although returns introduce significant noise into our model, there may be a relationship between recent returns, or squared returns, and future volatility. For instance, it is well-known that volatility is higher soon after negative returns than positive returns. Thus, it was important to include signed features to account for this hypothesis. We chose to include returns and squared returns over 1, 5, 30, 60, and 120 minute prior to a given prediction.

Metadata The Binance API provided the number of trades and amount of Bitcoin traded in USD, and our exploratory data analysis showed significant correlation between these features and future volatility. We expect these features to capture some information about market behaviour, independent of price moves. Both were included as sums over the past 30 minutes to obtain a stable estimate of trading activity. Similarly to volatility, we expected that a recent change in quote volume may indicate a change in trajectory, so we added the percent change in quote volume from t-60 to t-30 to t-30 to t.

Calendar and Time As per our observations about periodicity, we included time of day, and a binary flag to indicate whether or not a given day is a trading day in the US, UK, Germany, Japan, China, and Hong Kong. Germany was used as a proxy for Europe. Note that for training and prediction, we downsampled to 30-minute intervals, so the categorical variable time_of_day took 48 possible values.

Events We obtained data on fed events and CPI/employment data release dates. This was added as a categorical variable taking the value "NONE" if there was no event in the next 30 minutes, and "FED", "CPI", or "JOBS" for the other events.

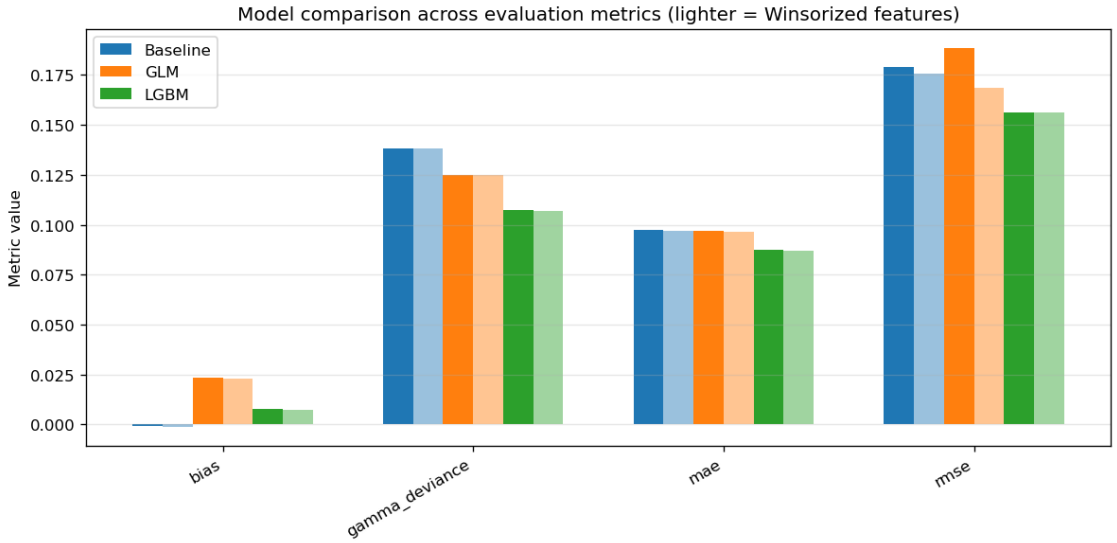
Model Training, Evaluation, and Performance

Training Metrics To train our models, we decided to use gamma deviance as our training metric, with increased sample weight for data points with FED/CPI/JOBS data releases. This weighting choice was made because these events only occurred about 60 times in about 40,000 samples. Gamma deviance was the correct metric to use for a strictly positive, right-skewed target. In addition, we added bias, MAE, and RMSE as secondary metrics in our final evaluation on the validation data set.

Hyperparameters In addition to the recommended hyperparameters in the assignment, we tested a log-transformation on strictly positive numerical features for the glm, as well as different values of `max_depth` for the lgbm (eventually deciding on 7).

To choose the best combination of these hyperparameters, we used grid search with cross-validation for the glm, and randomized search with cross-validation for the lgbm due to increased training time. Cross-validation was done with time series splits to prevent data leakage.

Results Below, we have a bar plot of gamma deviance and secondary metric for the glm, lgbm, and baseline model (50 minute ewm of squared returns).

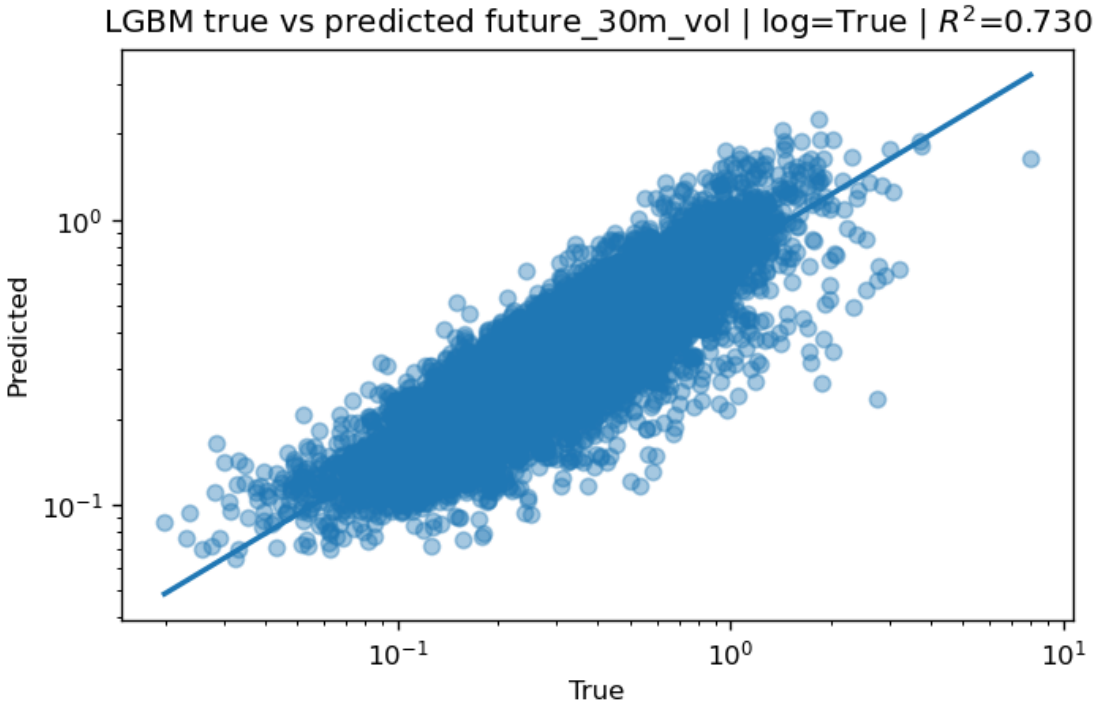


We make two observations. First, the lgbm outperforms both models on all metrics apart from bias, where baseline is the best. Secondly, Winsorized features improve the glm's rmse significantly, but has little impact otherwise. Thus, we selected the lgbm with Winsorized features as our final model, since it outperforms the raw features slightly.

Final Model Performance The lgbm with Winsorized predictors outperformed the baseline model with Winsorized predictors by 23% on gamma deviance, 10% on MAE, and 11% on RMSE, at the expense of increased bias. It is tough to say exactly how good this is without context, but it is definitely an improvement over both the glm and the naive baseline model.

To conclude this section, we plot predicted vs. actual 30 minute volatility in the validation set, with both axes log-scaled. For more visuals, including feature importance and partial dependence

plots, see the model_evaluation notebook.



0.1.4 Next Steps

Finally, we outline a few interesting ways in which our final model could be improved.

1. **More data sources**. As mentioned previously, Bitcoin interacts with other markets, such as equities, bonds, and indices. Including other features, such as S&P 500 returns or the volatility index (VIX), could capture more information than a purely bitcoin model. A more complete calendar of events that can impact the market, such as political speeches, would likely be helpful as well.
2. **Different features**. Our feature set was by no means complete. It could be interesting to add more lags of volatility, or a larger feature set that captures change rather than absolute values. Furthermore, due to the time constraint, we used the same feature set for the glm and lgbm. The glm especially would have benefited from more careful consideration of distributions of features.
3. **A different responder**. Especially for the glm, it may have been better to predict *change in volatility* rather than volatility itself. This would be closer to normally distributed, and woud likely be a better responder under the glm constraints.
4. **More time horizons**. This modeling approach could be extended to shorter or longer windows - it would be interesting to see how feature importance changes.