Why Machine Learning Models Often Underestimate Price Volatility

Yes, it's a frequently observed characteristic that for price prediction models trained using machine learning, the **magnitude of predicted prices is often smaller than the magnitude of actual prices**. This tendency is particularly pronounced in very **noisy (low signal-to-noise)** and **non-stationary environments**, such as financial markets.

This phenomenon isn't a flaw in the models per se, but rather a consequence of how they're trained and the inherent nature of predicting highly uncertain and volatile data. The core reasons can be broken down into several interconnected factors:

1. The Influence of the Loss Function: A Bias Against Large Errors

Most price prediction models are trained by minimizing a **loss function**, with **Mean Squared Error (MSE)** being one of the most common. MSE calculates the average of the squared differences between the predicted and actual values. By squaring the errors, larger mistakes are penalized exponentially more than smaller ones.

This has a significant side effect: to minimize the overall loss, the model is incentivized to be conservative. Making a very bold prediction that turns out to be wrong would result in a massive error, heavily impacting the total loss. Therefore, the model learns to "play it safe" by predicting values that are closer to the central tendency of the data it was trained on, inherently dampening the magnitude of its predictions to avoid potentially large penalties.

2. Regression Towards the Mean: A Statistical Reality

This is a fundamental statistical concept that machine learning models naturally embody. In essence, "regression towards the mean" states that extreme or outlier events are likely to be followed by more moderate ones. When a model is uncertain, its most rational guess is to predict a value closer to the average of what it has seen in the past.

In noisy and non-stationary price data, extreme price spikes or drops can be considered low-probability events. A machine learning model, trying to find a generalizable pattern, will often treat these extremes as noise rather than a repeatable signal. As a result, its predictions will be "pulled" back towards the mean, leading to an **underestimation of these volatile movements**.

3. The Challenge of Non-Stationarity and Low Signal-to-Noise Ratio

Financial markets are prime examples of **non-stationary** systems, where the statistical properties like mean and variance change over time. This makes it incredibly difficult for a model to learn stable, long-term patterns. When the underlying data-generating process is constantly shifting, the model struggles to distinguish between a genuine new trend and random noise.

In a **low signal-to-noise** environment, the true underlying price movement (the signal) is often obscured by a large amount of random fluctuation (the noise). A machine learning model, in its attempt to find a generalizable and robust pattern, will inherently learn to filter out what it perceives as noise. This filtering process often leads to a smoothing of the predictions, effectively reducing their magnitude compared to the more erratic actual prices.

4. The Dampening Effect of Regularization

Regularization techniques like **L1 (Lasso)** and **L2 (Ridge)** are commonly used to prevent a model from becoming overly complex and "overfitting" to the training data. They achieve this by adding a penalty to the loss function based on the size of the model's coefficients.

- **L2 Regularization** encourages smaller, more evenly distributed weights, which generally leads to more stable but also more conservative predictions.
- **L1 Regularization** can force the coefficients of less important features to zero, effectively simplifying the model.

Both methods, by discouraging large coefficients, can contribute to the model producing predictions with a smaller magnitude.

How Different Models Exhibit This Tendency

While this is a general trend, it can manifest differently across various types of machine learning models:

- **Linear Models:** These models are inherently limited in capturing the sharp, non-linear movements of volatile prices. Their predictions are, by their nature, more constrained and less likely to match extreme price swings.
- **Tree-Based Models (e.g., Random Forest, Gradient Boosting):** These models are excellent at finding complex patterns within the data they've seen. However, they struggle with **extrapolation**. A decision tree's prediction is based on the average of the data points in the "leaf" it lands on. If a future price is higher or lower than anything it has seen in the training data, the model's prediction will be capped at the maximum or minimum values it has learned. This makes them particularly prone to underestimating future volatility that exceeds historical levels.
- **Neural Networks (e.g., LSTMs, GRUs):** While more capable of capturing complex temporal dependencies and non-linearities, neural networks are not immune to the issues of loss function incentives and the influence of noisy data. They are also highly susceptible to overfitting in noisy environments, which necessitates the use of regularization techniques that can dampen prediction magnitudes. While they might be better at modeling volatility than tree-based models, they will still exhibit a conservative bias to avoid large prediction errors.

In conclusion, the **underestimation of price prediction magnitudes** by machine learning models is not a sign of failure but rather an intrinsic characteristic of their design and the challenging nature of forecasting in noisy, non-stationary domains. It reflects a model's attempt to be robust and generalizable in the face of significant uncertainty.