Alejandro Rodriguez
UCF ID: 5361434

Bitcoin Pricing Model Report

The objective of this assignment was to predict Bitcoin prices using a combination of chainlet data and historical price information. The model that was created has specified constraints like not being able to use price as our feature and ensuring no data leakage within our model which was primordial. This report will explain the model/feature selection/creation, the reasoning behind splitting the data the way we did, the models' performance which was calculated by the RMSE and the challenges that came with the assignment.

Feature Selection:

The first step to deciding on the features that were going to be used for the model was to create a correlation study. This showcased how much of our features were correlated to the price. It was evident that the data that we were using had very little correlation but a very high collinearity. This type of behavior is common when looking at financial data. This would mean that most of the data in the chainlet data set had very little predictive power for our model. To address the lack of strong predictive features we engineered new features like lag prices. From there we created 3 other features using the lagged prices which included moving averages, momentum and standard deviation. Momentum has been a metric that is known to capture the ability of an asset to keep the same movement (i.e. rising or crashing). Moving averages as a feature help address the outlier problem that the dataset might have. Finally, standard deviation was chosen as a metric to try to capture the volatility that Bitcoin has.

Model Selection and Peformance:

Our model selection process focused on two main categories: neural networks and linear regression approaches. We began by testing a neural network, but its performance was disappointing (with an RMSE around 2,000). We attribute this to both the limited size of our dataset and the fact that neural networks typically require larger volumes of data to effectively learn complex patterns.

We then explored various linear regression models, including Ordinary Least Squares (OLS), Lasso, Ridge, and Elastic Net. Among these, **OLS and Elastic Net emerged as the top-performing models**. OLS gave us a strong baseline, with an RMSE of approximately 1,048, likely due to the dataset's high collinearity. Elastic Net also performed best with an RMSE of 1,039, offering more flexibility by combining L1 and L2 penalties. However, we found the L1 component dominated in practice, suggesting that pure L2 (as in Ridge) did not contribute much to improving predictive power. Consequently, Ridge offered no notable advantage in this scenario, and Elastic Net behaved similarly to Lasso in terms of regularization.

Overall, the limited effectiveness of the neural network reinforced our belief that larger datasets are typically needed to leverage deep learning's strengths. Meanwhile, OLS and Elastic Net proved to be robust choices under our dataset's constraints, with Elastic Net's additional hyperparameters providing a valuable avenue for fine-tuning when compared to the other linear methods.

Splitting Data and Data Leakage

Our dataset was split into five equal folds using TimeSeriesSplit, which ensures that our training data is rotated sequentially so that the model does not use data past December 1st during training. We chose TimeSeriesSplit as our cross-validation technique because it preserves the chronological order of the data—a crucial factor when working with dates. Additionally, it allows us to set strict date parameters, ensuring that the model never incorporates future prices (e.g., those from December 2017) during training. This was critical for our model because if we didn't implement this the model would be predicting something that it has already seen, which goes against our objective.

Challenges

This project presented several challenges, starting with our limited initial knowledge of how to begin. My greatest difficulty was feeling overwhelmed by the scope of the work. To manage this, I broke the project into smaller, more manageable steps—first focusing on data cleaning, then on creating features that would support the model, and so forth. Adhering to the assignment's constraints was another obstacle, particularly because we were restricted in the features we could use. Relying on lagged price data meant the model struggled to stay ahead of real-time price fluctuations. Furthermore, our options for modeling were limited to linear methods and neural networks. Given the data's significant covariance and collinearity, it was challenging for these approaches to accurately capture Bitcoin's somewhat random walk behavior.

Local Explanation with LIME

The latest implementation to the project was the addition of Local Interpretable Model-Agnostic Explanation (LIME) as our explainable AI agent. The introduction of this gives us an insight to how our model is performing and returns the features that has the highest importance in our predictions for December. The three days chosen to explore were December 1st, 15th, and 31st.

December 1, 2017

- Key Positive Features:
  - price_lag_1: This indicated that the previous day's price had a strong influence on raising the predicted price for December 1.
  - ma_5: The 5-day moving average also contributed positively, suggesting that the recent upward trend carried weight.
- Key Negative Features:
  - Certain chainlet features (e.g., 5:2_occ or 2:10_amo if they appeared) slightly pulled the prediction downward, implying that some transaction activity signals did not support as high a price on this day.

December 15, 2017

- Key Positive Features:
  - momentum: The difference between the previous two days' prices showed a continued bullish trend, lifting the model's prediction.
- Key Negative Features:
  - Some chainlet features indicated reduced activity or amounts, which slightly dampened the model's predicted price.

Overall, the LIME explanation suggests the model was heavily influenced by recent momentum indicators and short-term moving averages, while the chainlet data in this instance had a smaller effect.

December 31, 2017

- Key Positive Features:
    - Again, price_lag_1 and the short window moving averages were the strongest indicators driving price upward.
- Key Negative Features:
    - A handful of chainlet occurrence features (x:y_occ) had modest negative weights, indicating they expected a slightly lower price.

Overall, LIME shows that historical price-based signals continued to dominate the model's local decision at the end of December.

The most notable features for our predictions were lagged prices and moving averages. The chainlets were present as positive features in some days but most of the days they had smaller weights than the aforementioned. This outcome aligns with our earlier statement that shows that the strongest signals for our models were historical price driven. Overall, the usage of LIME gives us insight into the black box nature of models by showing us what features influenced the predictions the most.