Crypto Forecasting Using Machine Learning

Abstract

The popularity of blockchain and cryptocurrencies has significantly increased in recent years, primarily due to their rising trading volumes and large market capitalization. Cryptocurrencies are no longer limited to trading, but are also being accepted for monetary transactions. As the return on investment rises with fluctuating prices, there is an increased interest among investors, traders, and the general public in Bitcoin and other crypto currencies. This study aims to implement forecasting models that accurately predict changes in cryptocurrency prices. Bitcoin, Ethereum, and Litecoin prices were predicted using a gradient boosted decision tree and a temporal neural network mixed with LSTM.

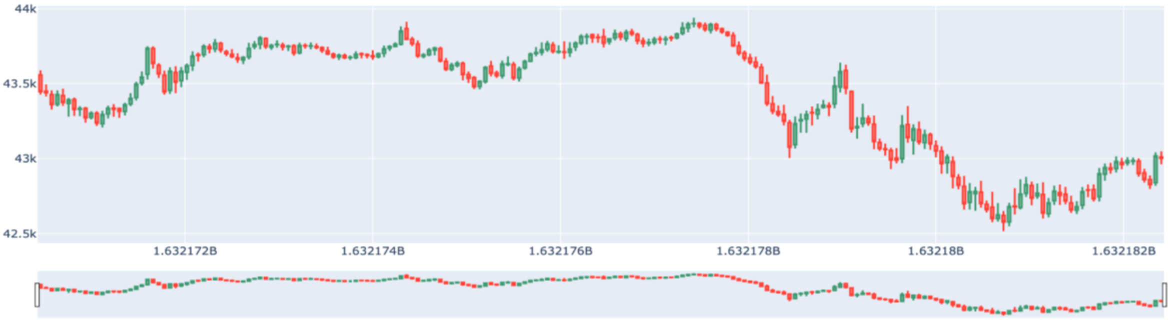
Dataset:

The Daily historical data of the crypto market was taken from Kaggle which included 10 currencies including Bitcoin, Ethereum, etc. The data used is from 1st Jan 2020 to 1st Dec 2020 to give a large amount of data while still having the computational support to compute it.The dataset contains minute-by-minute data of different cryptocurrencies, represented by their asset ID (e.g., Asset_ID = 1 for Bitcoin), with timestamps provided as Unix timestamps (the number of seconds elapsed since 1970-01-01 00:00:00.000 UTC). The dataset includes various parameters such as the total number of trades in the time interval, opening price, highest price reached, lowest price reached, closing price, quantity of asset bought or sold displayed in the base currency USD, and VWAP (the average price of the asset over the time interval, weighted by volume).

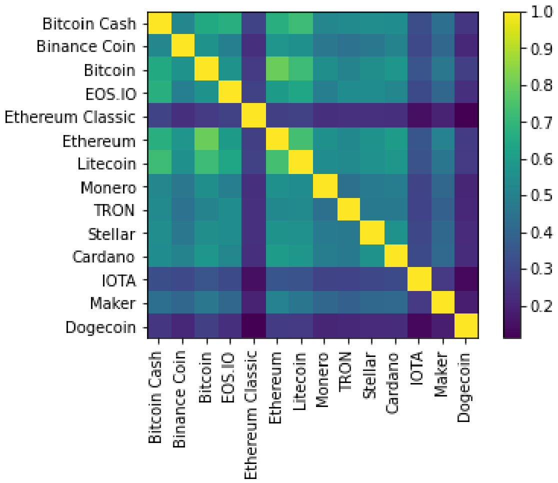
Additionally, the dataset includes the target variable, which represents the residual log-returns for the asset over a 15-minute horizon. The residual log-returns indicate the difference between the actual log-returns and the predicted log-returns for a given time interval and are used to evaluate the accuracy of forecasting models. This dataset was used to build predictive models for cryptocurrency price movements based on the provided features and the target variable.

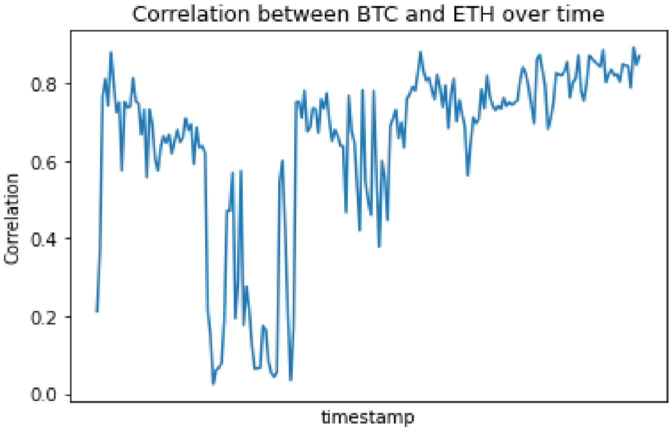
Data visualization:

The data was a mixture of all the different cryptocurrencies in one data frame. So firstly, we split the dataset in separate data frames depending on the asset (i.e., Currency). Further we visualized our data by using a candlestick chart. Below is an example of the candlestick chart of the bitcoin crypto.



The crypto market contains coins which share a range of correlation that converge and diverge as time changes. Understanding the interaction between crypto coins might provide insight into the market as a whole as some cryptos might be following the trends of the giants including bitcoin.





The correlation Matrix above shows the correlation between all the cryptocurrencies. We could see that Ethereum and bitcoin had the highest correlation. Which made sense, as they are two biggest cryptocurrencies out there.

Pre-processing:

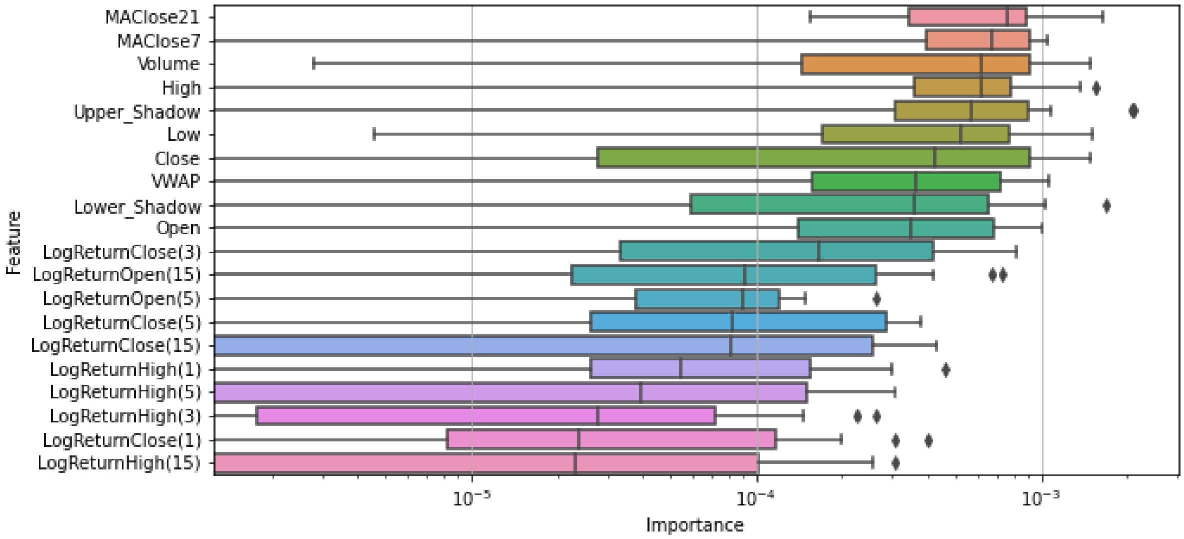
Dealing with missing data- In this dataset, missing data for a particular cryptocurrency asset at a certain minute is not denoted by NaN values, but rather by the absence of corresponding rows. We can see this by checking the timestamp difference between consecutive rows to see if there is missing data. In order to deal with these missing values, we replace them with the previous valid value.

When analyzing price changes for an asset, it is common to use the price difference. However, comparing returns across different assets with different price scales can be challenging. We tackled the issue of comparing returns across different assets with varying price scales. We found that computing the percentage change in price, also known as the return, was an effective solution. However, to model time series mathematically, we chose to use log returns instead of regular returns.

Log returns are preferred because they are additive across time and unbounded, while regular returns have a lower bound of -100%. To calculate the log return, we simply took the logarithm of the ratio between two consecutive prices. We noticed that the first row of the data had an empty return value since the previous price was unknown. To address this, we simply dropped the empty return data point. This allowed us to accurately compare returns across different assets and model time series data effectively.

Approach:

To begin with, we developed a decision tree model for the entire cryptocurrency market. Then, we used this model to perform feature selection and modeling, which we applied to our more specialized models developed for individual cryptocurrencies. One of the key features we engineered was the LogReturn between different time periods, and we also used moving averages to identify general trends. To improve our understanding of feature importance and reduce noise, we employed cross-validation techniques to run the decision tree model multiple times. We used cross validation to run our general decision tree model multiple times to remove some noise and get a better view of feature importance.



At first, the decision tree exhibited high performance on the training dataset but not on the testing dataset, indicating that the model was overfitting. To address this issue, we adjusted various parameters such as the depth of the tree and introduced regularization techniques to help improve the model's generalization capabilities.

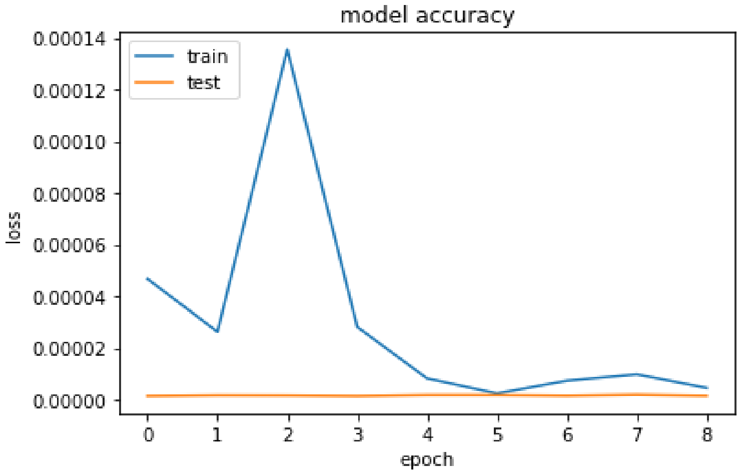
Given that the stock market data is in the form of time series, we attempted to construct an LSTM model to fit onto the data. We also explored the possibility of implementing a mixed TCN and LSTM model with the aim of treating various cryptocurrencies in a temporal manner. However, due to insufficient computational resources, we were unable to apply this approach to all cryptocurrencies and limited our focus to Bitcoin. Consequently, we adopted the TCN-LSTM model, which can learn from both directions of data flow. Our intention was to enhance accuracy and address the vanishing gradient problem commonly encountered with LSTMs.

After having selected the top features, we then focused on Bitcoin to try to build a model that has some predictive power. We managed with moderate success with creating a decision tree. To measure the performance of the model we compared whether the model predicted a possible return when that was actually the case or a negative return when that was actually the case.

Results:

In comparison to the Boosted Decision Tree, the TCN-LSTM model yielded slightly better results, although neither performed exceptionally well on the crypto market. With a window size of 30, the TCN-LSTM model achieved an accuracy of 51%, whereas the TCN-LSTM model with a window size of 50 failed to yield any significant improvement, also achieving an accuracy of 51%. The Boosted Decision Tree model did not perform well, producing a Pearson correlation of only 0.023 and resulting in an accuracy rate of 51.2% for its predictions. Overall, this was a challenging problem with no significant breakthroughs in terms of results.

The below chart represents one of the training cycles for the TCN-LSTM. It can be noted that the validation loss barely changes which represent having a model with very low predictive power.



Distribution of work:

Risa Fukutoku (54321338): Was in charge of data preparation, visualizing features, and preprocessing.
Mathias Moelgaard (47929074): Was in charge of building the TCN-LSTM and Gradient Boosted Decision Tree with Regression.
Yash Sawant (533687872): Was in charge of reading feature importance, overfitting, and parameter tweaking.

Reference

Alessandro Ticchi, Andrew Scherer, Carla McIntyre, Carlos Stein N Brito, Derek Snow, Develra, dstern, James Colless, Kieran Garvey, Maggie, Maria Perez Ortiz, Ryan Lynch, Sohier Dane. (2021). G-Research Crypto Forecasting. Retrieved March 21, 2023 from https://kaggle.com/competitions/g-research-crypto-forecasting.

Andrew Scherer, Carla McIntyre, Carlos Stein N Brito, Derek Snow, dstern, James Colless, Kieran Garvey, Maria Perez Ortiz, Ryan Lynch, Sohier Dane, Tom Van de Wiele. (October 2021). G-Research Crypto Forecasting, Version 9. Retrieved March 21, 2023 from https://www.kaggle.com/code/cstein06/tutorial-to-the-g-research-crypto-competition/notebook.