Kaggle: G-Research Crypto Forecasting

Li Chenghai(20828446): Coding Support and Report Writing
Liu Shifei(20734530): Report Writing and PPT
Nie Jialei(20747874): Coding Support, Report Writing, PPT and Presentation
Wu Jiajun(20666111): Coding, Report Writing, PPT and Presentation
Dec. 10, 2021

I. Introduction

Over $40 billion worth of cryptocurrencies are traded every day. They are among the most popular assets for speculation and investment, but they have proven wildly volatile. Fast-fluctuating prices carry huge risks, so we try to predict price movements in advance.

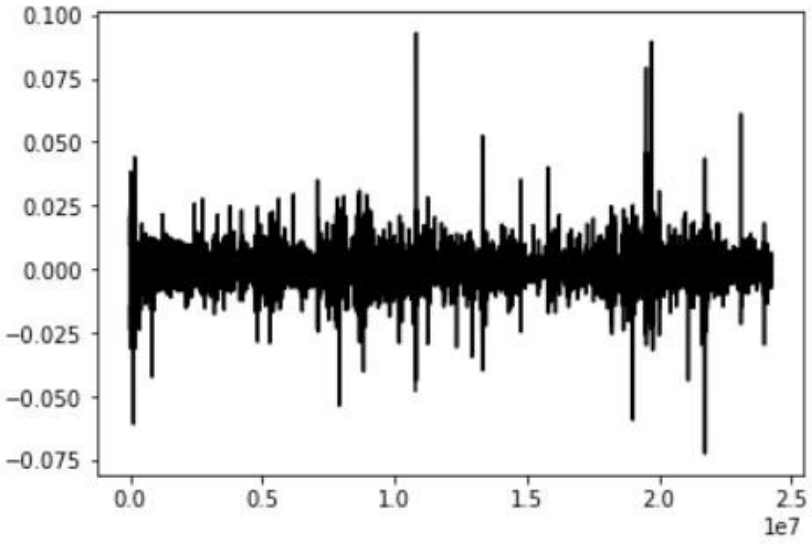
In this competition, we used machine learning to forecast short term returns in 14 popular cryptocurrencies, such as Bitcoin, Binance Coin, Monero and so on.

II. Data Description

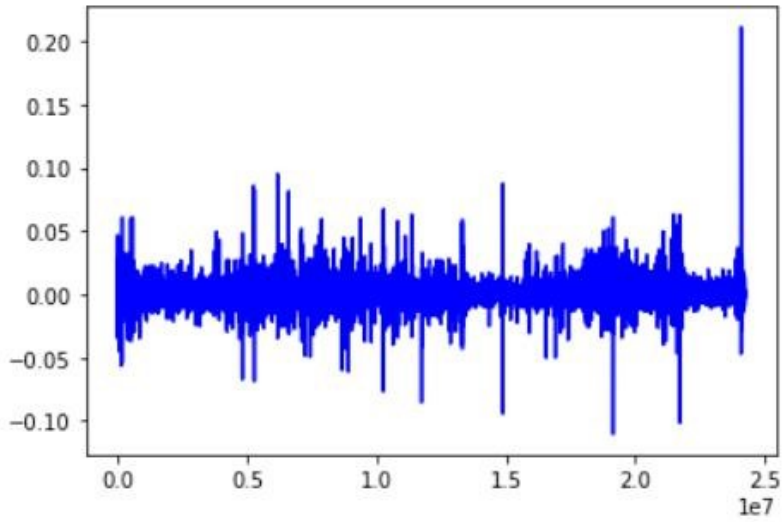
There is a dataset of millions of rows of high-frequency market data dating back to 2018 which we used to build the model. In the whole calculation process, we used three data sets, namely train.csv, example_test.csv and asset_details.csv.

The training set contains information such as timestamp, open price, close price, high price, low price and Target. The example testing set is an example of the data that will be delivered by the time series API. The asset details set provides the real name of the cryptoasset for each Asset_ID and the weight each cryptoasset receives in the metric. This weight represents our investment ratio in each cryptocurrency when we invest in cryptocurrencies.

We drew the Target sequence of Bitcoin into the following image. As we can see from the figure below, Bitcoin's return is in a state of high volatility, with the highest being 0.09 and the lowest being less than -0.07. However, we can find that most of the returns are still concentrated between 0.025 and -0.025.



And we drew the Target sequence of Litecoin into the following image. As we can see from the figure below, Litecoin's return is also in a state of high volatility, with the highest being higher than 0.2 and the lowest being less than -0.1. Similarly, we can find that most of the returns are still concentrated between 0.05 and -0.05. Compared with Bitcoin, Litecoin's return is in the normal fluctuation range most of the time, and only occasionally there are extremely large return fluctuations.



III. Methodology

(1) Model - LightGBM

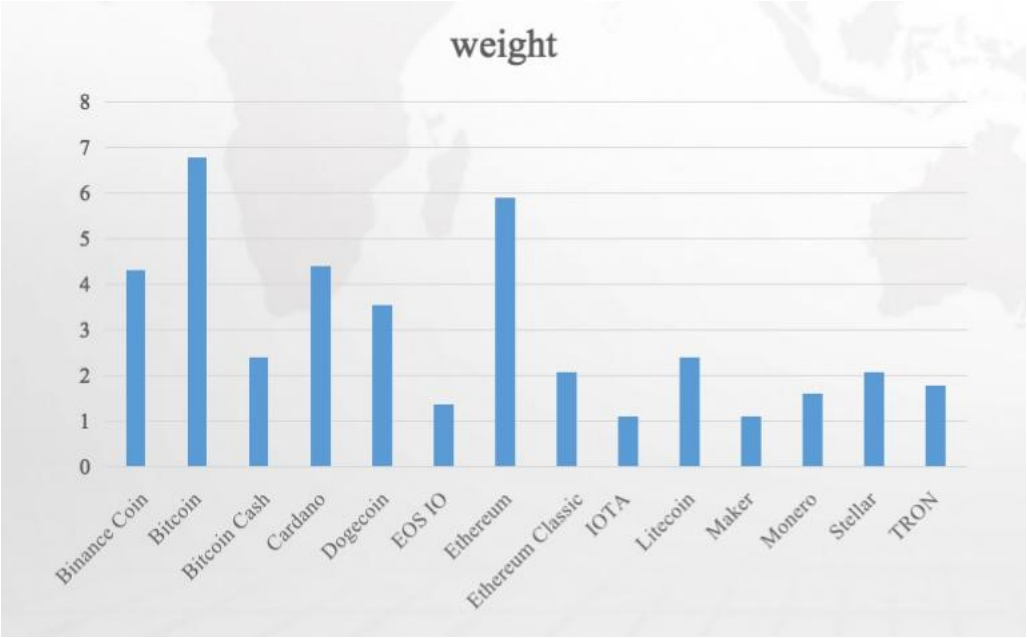
GBDT (Gradient Boosting Decision Tree) is a commonly used model in machine learning. Its main idea is to use weak classifiers (decision trees) to iteratively train to obtain the optimal model. The model has the advantages of good training effect and not to overfit. And LightGBM (Light Gradient Boosting Machine) is a framework that implements the GBDT algorithm. It supports high-efficiency parallel training, and has faster training speed, lower memory consumption and better accuracy. It also supports for distributed, and can quickly process massive amounts data.

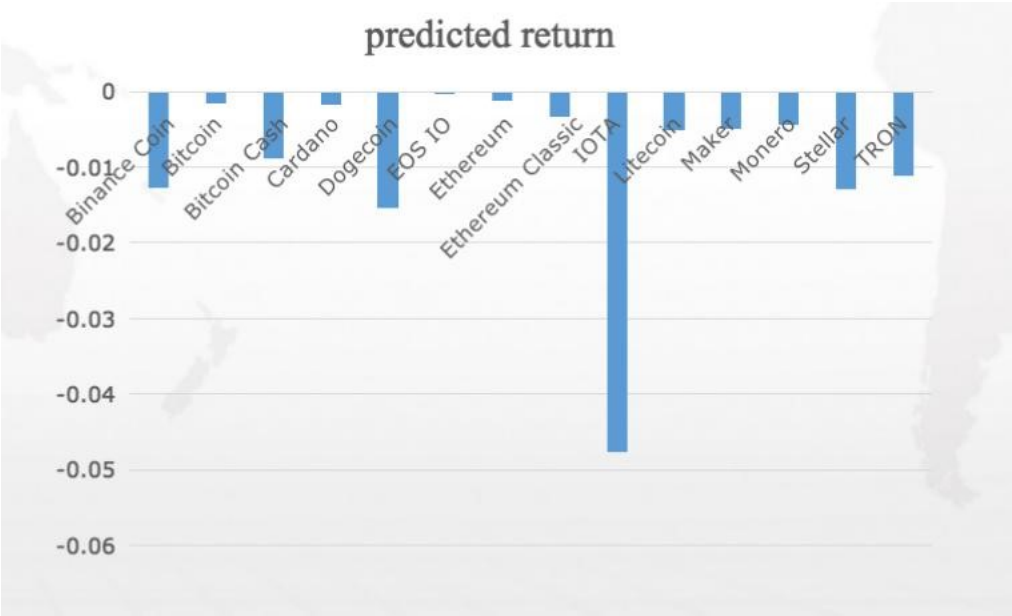
(2) Analysis procedure

There are several steps of the modeling process. First of all, make the data only the latest one. Remove the data that becomes leak and create valid data. The second step is to make feature function. The third step is training and validation. After training, we need first save the model. Then we calculate total validation score, not per crypto coin. The final step is to make prediction.

IV. Prediction

The following shows the predicted log return for 14 different cryptocurrencies at future time point (1623542580).


weight
8
7
6
5
4
3
2
1
0
Binance Coin
Bitcoin
Bitcoin Cash
Cardano
Dogecoin
EOS IO
Ethereum
Ethereum Classic
IOTA
Litecoin
Maker
Monero
Stellar
TRON


predicted return
0
-0.01
-0.02
-0.03
-0.04
-0.05
-0.06
Binance Coin
Bitcoin
Bitcoin Cash
Cardano
Dogecoin
EOS IO
Ethereum
Ethereum Classic
IOTA
Litecoin
Maker
Monero
Stellar
TRON

Appendix

LightGBM is a GBDT based model, which involves two effective techniques to significantly decrease the total computation cost. And to some extent, by empirical analysis, these techniques can also play a role to reduce over-fit, thus help the model achieve a robust result.

(1) Gradient-based One-Side Sampling (GOSS).
Here we select the data by the training loss rather than using all the data points during the iterations. The data points which have high prediction errors will be paid more attention to. Thus, the model will select the points with top α percent, and then randomly select β percent of total data points in the remaining data. To rebalance the selection bias, a scale $\frac{1-\alpha}{\beta}$ with is larger than 1 will be multiplied to the gradient of the β percent of data points. Because of $\alpha + \beta < 1$, the model needs to take care of much fewer data points when calculating gradients. And this technique indeed plays a similar with Dropout, which is proven to slow down overfitting in deep learning.

(2) Exclusive Feature Bundling (EFB).
In many scenarios, the input features can be very sparse. The model leverages a greedy algorithm to determine which features are mutually exclusive, thus we can bundle them with minimal loss. The paper proves that if two features never take nonzero values simultaneously, we can fuse them without losing information. And we can become more aggressive when we want to achieve a higher compression ratio by bundle features that possess low conflict counts.

(3) Bucketing.
The split algorithm will rely on a pre-sort result to find the optimal split points. Due to the iteration loop, this can be a time-consuming procedure, in which each loop has s O(#data × #feature) difficulties. The bucketing is to reduce the number of points by merging the neighboring points. The computation complexity can turn into O(#bin × #feature). The other frameworks such as XGBoost also adopt this technique.