G-Research Crypto Forecasting

Cao Bokai,Luo Zhuang,Shi Jie,Lai Fujie

December 12, 2021

Abstract

Over $40 billion worth of cryptocurrencies are traded every day. They are among the most popular assets for speculation and investment, yet have proven wildly volatile. Fast-fluctuating prices have made millionaires of a lucky few, and delivered crushing losses to others. In this competition, we machine learning expertise to forecast short term returns in some popular cryptocurrencies. In this report, we mainly focus on five parts. First is the data preparation. Second is feature engineering, where we design lots of features. The third part is brief introduction to our model and method used. After that, we give performance evaluation and interpretation of the result based on our model. We make our conclusion and prediction of the cryptocurrencies return trend.

Workload: Cao Bokai finished the feature engineering part and participated in model design as well as wrote corresponding report. LaiFujie participates in model selection and partly model building and training and also writing report of LSTM. Luo Zhuang did most of the coding and report composing work, and wrote the LightGBM part of the report. Shi jie finished data preparation part and made the feature engineering table.

Part I: Data preparation

We use time series data of 14 cryptocurrencies including Bitcoin Cash, Binance Coin, Bitcoin, EOS.IO, Ethereum Classic, Ethereum, Litecoin, Monero, TRON, Stellar, Cardano, IOTA, Maker, Dogecoin, 5 datasets including asset_details.csv, example_sample_submission.csv, example_test.csv, supplemental_train.csv, train.csv. train.csv has 10 columnsincluding timestamp, Asset_ID, Count, Open, High, Low, Close, Volume, VWAP, Target. These data dating back to 2018, and each row of dataset contains transaction information in 1 minute, we also performed some simple processing on the data, such as deleting outliers.

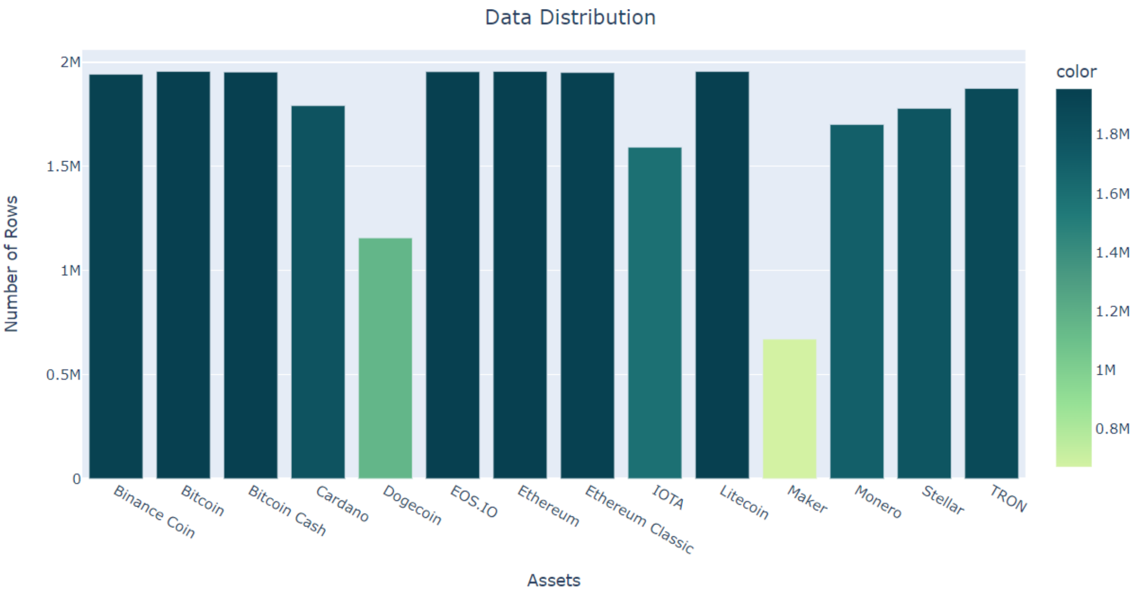

Figure 1: Data Distribution of Training Data.

Part II: Feature Engineering

Feature engineering is of vital importance that influenced model performance directly. The raw data only contains basic price and volume like HIGH, LOW, VWAP… It may be difficult for models to study patterns using these columns. So, it's necessary to add columns. At first, we made some factors with practical financial significance by simple transformation using price and volume data. Like spread(HIGH-LOW), upper/lower shadow, MOV, CLS…

Besides above factors and basic price-volume data, we also take some outstanding alphas in conventional financial market (e.g. stock, futures market) as a reference. We made the variable selection since some of them may not make sense. Fundamental factors and factors including lateral ranking functions are dropped. WorldQuant Alpha101[1], GuoTaiJunAn Alpha191, and some other alpha factors are used in our model. This kind of alpha factors are interpretations of crypto market inherent law, might help us improve sharpe ratio and return. As feature engineering work, our factors will directly promote the model performance to a more precise, robust, comprehensible one. For example, Alpha002 in the code, (-1*DELTA((((CLOSE-LOW) - (HIGH-CLOSE)/(HIGH-LOW)),1)), indicates long short power imbalance and how it moves since ((CLOSE-LOW)-(HIGH-CLOSE))/(HIGH-LOW) is clearly unbalancedness of long and short. Alpha013 (((HIGH*LOW)0̂.5)-VWAP) is inverse type factor while

[1] Zura K , Geoffrey L , Igor T . 101 Formulaic Alphas[J]. Ssrn Electronic Journal, 2015.

Table 1: List of Factors

Factor	Describe	Meaning
VWAP		Volume-Weighted Average Price
speard	High - Low	The highest price minus the lowest price during the minute
mean_trade	Volume / Count	The number of cryptosaaet u units traded per trade during the minute
log_price_change	log(Close /Open)	The log of quotient of close price divided by open price
upper_Shadow	High - max(Close, Open)	
lower_Shadow	min(Close, Open) - Low	
high_div_low	High / Low	The quotient of high price and low price
trade	Close - Open	The difference between close price and open price
gtrade	trade / Count	The quotient of trade and count
shadow1	trade / Volume	The quotient of trade and volume
shadow3	upper_Shadow / Volume	The quotient of upper shadow and volume
shadow5	lower_Shadow / Volume	The quotient of lower_Shadow and volume
diff1	Volume - Count	The difference of Volume and Count
mean1	(shadow5 + shadow3) / 2	The mean of shadow3 and shadow5
mean2	(shadow1 + Volume) / 2	The mean of shadow1 and volume
mean3	(trade + gtrade) / 2	The mean of trade and gtrade
mean4	(diff1 + upper_Shadow) / 2	The mean of diff1 and upper_shadow
mean5	(diff1 + lower_Shadow) / 2	The mean of diff1 and lower_shadow
UPS	High - max(Close,Open)	Upper shadow
LOS	min(Close, Open) - Low	Lower shadow
RNG	(High - Low) / VWAP	The quotient of speard and VWAP
MOV	(Close - Open) / VWAP	The quotient of trade and VWAP
CLS	(Close - VWAP) / VWAP	The change rate of close and VWAP
LOGVOL	log(1. + Volume)	Turn Volume into log value
LOGCNT	log(1. + Count)	Trun Count into log value
Close/Open	Close / Open	The quotient of close and open
Close-Open	Close - Open	The difference between close price and open price
High-Low	High - Low	The difference between high and low
High/Low	High / Low	The quotient of high price and low price
Mean	The mean of Open, High, Low, Close	
High/Mean	High / Mean	The quotient of high price and Mean
Low/Mean	Low / Mean	The quotient of low price and Mean
Volume/Count	Volume / (Count + 1)	
mean_price	The mean of Open, High, Low, Close	
median_price	The median of Open, High, Low, Close	
high2mean	High / mean_price	The quotient of High and mean_price
low2mean	Low / mean_price	The quotient of Low and mean_price
high2median	High / median_price	The quotient of High and median_price
low2median	Low / median_price	The quotient of Low and median_price
volume2count	Volume / (Count + 1)	

momentum factor Alpha014, (CLOSE-DELAY(CLOSE,5)). Both alpha factors are written together as a class.

Part III: Model Selection

In the G-Research competition, our goal is to process cryptocurrency's time series data and do price predictions. In traditional finance research, ARIMA and GARCH are the most useful models and they are built with clear assumptions on time series data, which allows them strong explanatory ability. But the problem is also the assumption that time-series data have the stability to some extent. We can make the raw price data stable using log return or MinMaxScaler for sure but in this way, we will lose some information hiding in original data. So some research has been done and one of them suggests that LSTM outperforms traditional-based algorithms such as the ARIMA model. More specifically, the average reduction in error rates obtained by LSTM was between 84 - 87 percent when compared to ARIMA indicating the superiority of LSTM to ARIMA[2].That's the reason for us to choose LSTM as our first model.

LSTM

LSTM refers to long short-term memory, is one of the recurrent network structures. A common LSTM unit is composed of a cell, an input gate, an output gate and a forget gate. The key to LSTMs is the cell state. The cell state is kind of like a conveyor belt. It runs straight down the entire chain, with only some minor linear interactions. It's very easy for information to just flow along with it unchanged. The cell remembers values over arbitrary time intervals and the three gates regulate the flow of information into and out of the cell. Gates are a way to optionally let information through. They are composed out of a sigmoid neural net layer and a pointwise multiplication operation. By these structures, LSTM can be trained to memory or forget certain information.

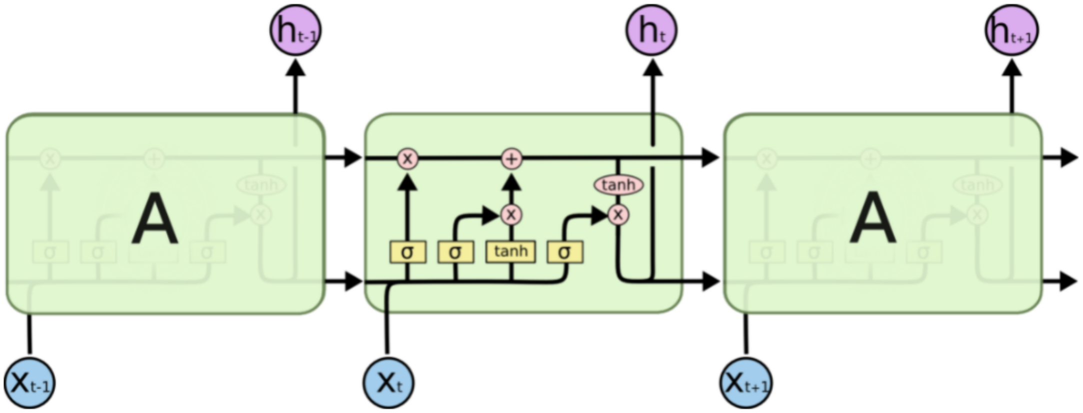

Figure 2: The repeating module in an LSTM contains four interacting layers.

Another famous model is Transformer. With its attention structure, the model may perform better than LSTM in many areas. However, because attention or bi-direction LSTM will consider future information in our cryptocurrency's case, we will not use those models.

[2] A Comparison of ARIMA and LSTM in Forecasting Time Series,Sima Siami-Namini; Neda Tavakoli; Akbar Siami Namin,2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA).

LSTM Workflow

Because our object is to predict 15min future return, so we set out window size to 15 and train for 10 epochs. After 10 epochs the training loss and validation loss decrease very slowly so the 10 epochs are for accelerating our training. After data preprocessing and feature engineering, we divide our dataset into ninety percent training data and ten percent testing data and set batch size as 1024. The input will have tensor shape(batch size, window size, number of cryptos, feature num) and the target will have shape(batch size, number of cryptos). We use the Adam optimization algorithm and set the learning rate as 0.001. For the loss function, we compute the cosine similarity between labels and predictions.

We first divide the dataset based on different cryptocurrencies. For each cell in the LSTM layer, we set a hidden state vector with size 32. After putting data into the LSTM network we do global average pooling for the output so that we can get the average of overall sequences. We do these for each cryptocurrency. Second, we concat all the output data and put them into a 128 units linear layer, and finally get the output.

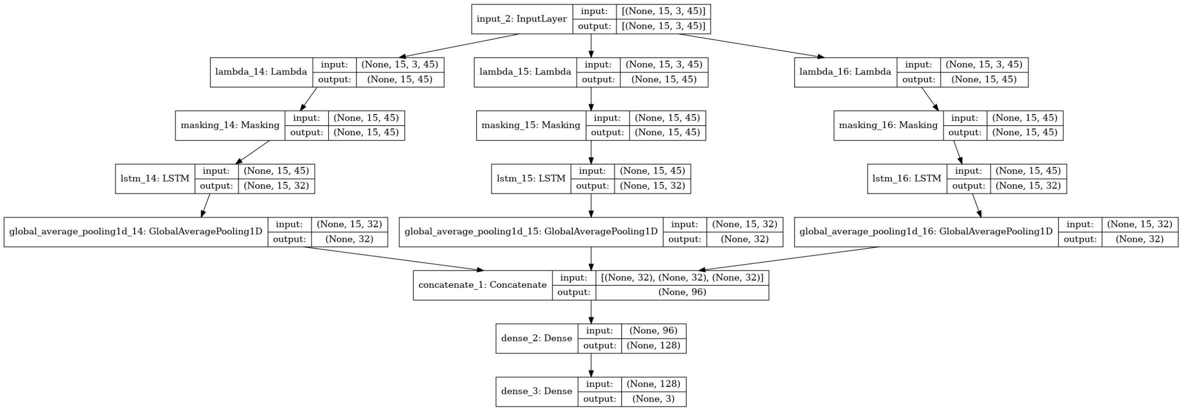

Figure 3: LSTM model

LSTM Performance

Since the test set was not given in the competition, we randomly selected 90% of the given train.csv for training and the remaining 10% for validation to judge the effectiveness of the model. In the competition, our predictions will be evaluated on a weighted version of the Pearson correlation coefficient, with weights given by the weight column in the asset details file. So we calculate the correlation of our prediction values and true values and we get the Pearson value for each asset, shown below.

Table 2: LSTM's Correlation of Validation Data

Coin	Cardano	Bitcoin Cash	Binance Coin	Bitcoin	Dogecoin	EOS.IO	IOTA
Coef	0.0254	-0.0063	0.0052	0.009	-0.0009	0.0126	0.0024
Coin	Ethereum	Ethereum Classic	Litecoin	Maker	TRON	Stellar	Monero
Coef	0.0155	0.0111	0.0009	0.0105	0.0326	0.0070	-0.0123

One of the biggest problems we have met in this competition is overfitting. Even we increase training epochs the loss in the validation dataset won't decrease continuously as we can partly see in the first ten epochs training. Other researchers' work also proves that increasing the number of epochs won't lead to improvement of prediction performance. It was noticed that in their research the number of training times or epochs had no effect on the performance of the trained forecast model and it exhibited a truly random behavior.

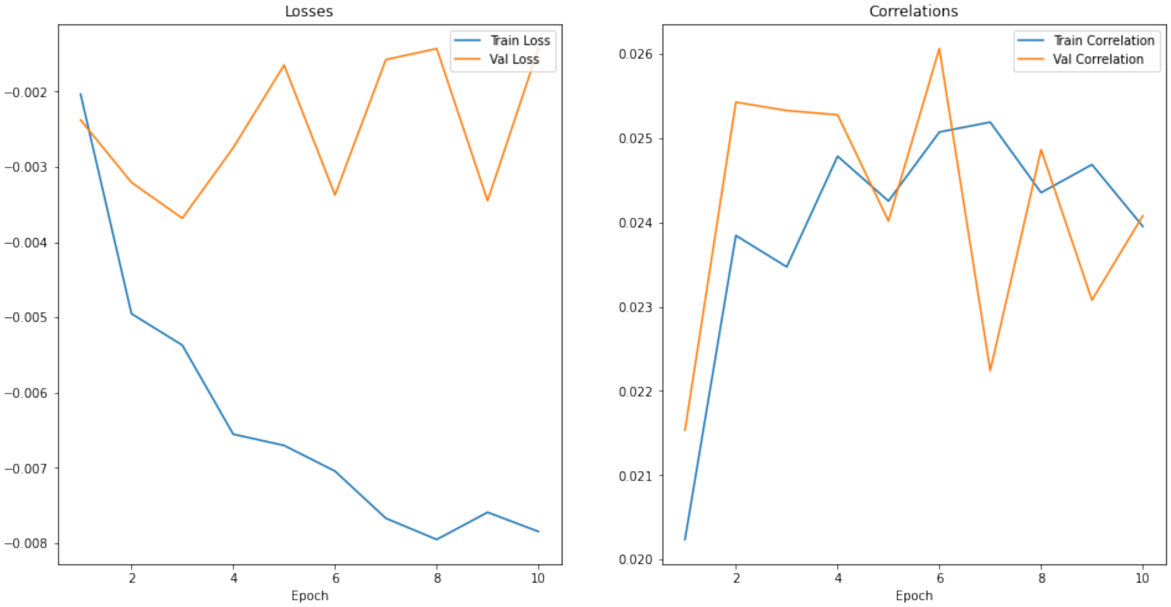

Figure 4: Loss & Correlation of LSTM model

LightGBM

LightGBM is a fast, distributed, high performance gradient boosting framework based on decision tree algorithms open-sources by Microsoft. It uses a histogram-based algorithm to speed up the training process, reduce memory consumption and combine advanced network communication to optimize parallel learning, called parallel voting DT algorithm. Also, LightGBM uses the leaf-wise strategy to grow trees and find a leaf with the largest gain of variance to do the split. The library is used extensively in Kaggle competitions, and often forms part of the winning solution. Other famous gradient boosting tree like Xgboost is slightly different from lightGBM in some algorithm details,but overall they share the same idea of gradient boosting and minimize the loss function.

LightGBM Workflow

Similarly to LSTM, we split train.csv to judge the model effect. We use the previously constructed factors to train the training set and use grid search for tuning the important parameter learning rate, feature_fraction, num_leaves and max_depth.

LightGBM Performance

For the constructed factors, LightGBM can output the importance of the features, sorted in descending order as follows. It can be seen that Mean(Mean of ’Open’, ’High’, ’Low’, ’Close’) and LOGCNT(Trun Count into log value) are important for predicting cryptocurrency returns.

We also calculate the correlation between the predicted and true values on the validation set and obtain the Pearson values for each asset as shown in the table below. It can be seen that for the same set of factors, LightGBM performs better on the validation set compared to LSTM.

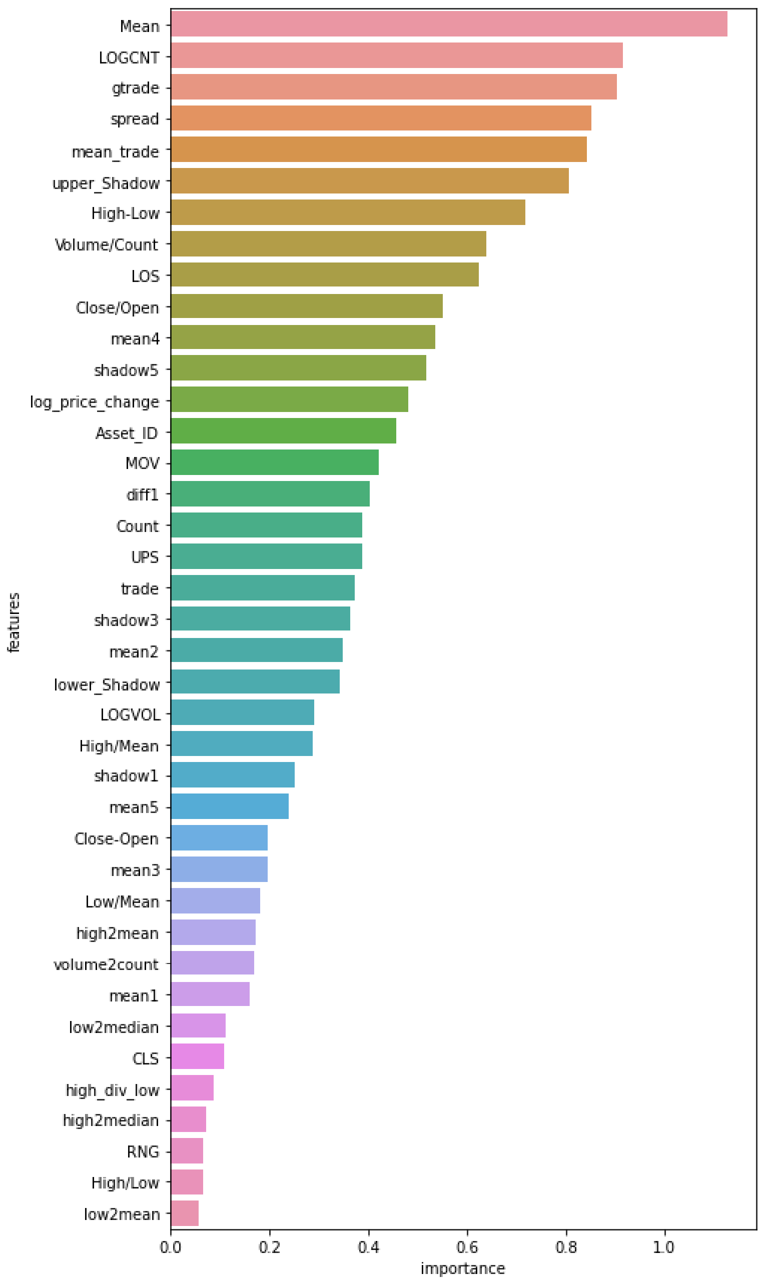

Figure 5: LightGBM Features Importance

Table 3: LightGBM's Correlation of Validation Data

Coin	Cardano	Bitcoin Cash	Binance Coin	Bitcoin	Dogecoin	EOS.IO	IOTA
Coef	0.0481	-0.0014	0.0174	-0.0018	0.0638	0.0006	-0.0029
Coin	Ethereum	Ethereum Classic	Litecoin	Maker	TRON	Stellar	Monero
Coef	0.0062	0.0104	-0.0331	0.0045	-0.0140	-0.0092	-0.0028