G-Research Crypto Forecasting

MAFS 6010Z Project 3

Wang Lei, Wang Zhongchen, Ye Xiaoyu and Zhang Quandi

{lwangcu, zwangfz, xyeak, qzhangcd}@connect.ust.hk

Department of Mathematics, HKUST

1. Introduction

This project works on the Kaggle contest "G-Research Crypto Forecasting", aiming to predict highly-frequency and real crypto market data. Two machine leaning techniques are applied: Linear Model and XGBoost. To solve the problem of volatility clustering, time series model and Box-Cox transformation are used to refine the models. Following the instruction of the contest, we adopt the weighted correlation as the indicator to evaluate model performance.

2. Dataset description

The dataset contains 14 actively traded cryptocurrencies with their 1-min trading data from 2018 to September 2021 and there are 24,236,806 history data of the cryptos in total. The training data range from the beginning of each asset to the end of 2020, and test data range from the beginning of 2021 to the end of the data (i.e. 2021-09-21). We fill in each of the missing value with previous non-missing value by using pad method.

Details of the data structure is shown in Table1. Note that the column "Weight" assigns a weight to each cryptocurrency asset for calculating the weighted correlation in the final evaluation step.

Table 1 - Basic Information of Data by Asset Class

Asset ID	Asset Name	Weight	Start Time	End Time
0	Binance Coin	4.304065	2018-01-01T00:01:00	2021-09-21T00:00:00
1	Bitcoin	6.779922	2018-01-01T00:01:00	2021-09-21T00:00:00
2	Bitcoin Cash	2.397895	2018-01-01T00:01:00	2021-09-21T00:00:00
3	Cardano	4.406719	2018-04-17T09:11:00	2021-09-21T00:00:00
4	Dogecoin	3.555348	2019-04-12T14:34:00	2021-09-21T00:00:00
5	EOS.IO	1.386294	2018-01-01T00:01:00	2021-09-21T00:00:00
6	Ethereum	5.894403	2018-01-01T00:01:00	2021-09-21T00:00:00
7	Ethereum Classic	2.079442	2018-01-01T00:01:00	2021-09-21T00:00:00
8	IOTA	1.098612	2018-05-09T08:07:00	2021-09-21T00:00:00
9	Litecoin	2.397895	2018-01-01T00:01:00	2021-09-21T00:00:00
10	Maker	1.098612	2018-05-10T15:21:00	2021-09-21T00:00:00
11	Monero	1.609438	2018-01-01T00:01:00	2021-09-21T00:00:00
12	Stellar	2.079442	2018-02-16T23:53:00	2021-09-21T00:00:00
13	TRON	1.791759	2018-02-06T21:37:00	2021-09-21T00:00:00

In the train data, number of trades, open-high-low-close prices, trading volumes, volume-weighted average prices and the target to predict (returns) are provided for each 1-min time interval.

3. Exploratory data analysis

To capture some main characteristics of the data, we firstly visualize the close price, and the log returns over the whole period of the 14 cryptocurrencies (Figure1 & 2). Most of the cryptocurrencies demonstrate a surge in price and a more volatile change after year 2021, except for Bitcoin Cash and EOS.IO, indicating a significant style change of crypto market.

Figure 1 - Close Prices of 14 Cryptocurrencies Over Time

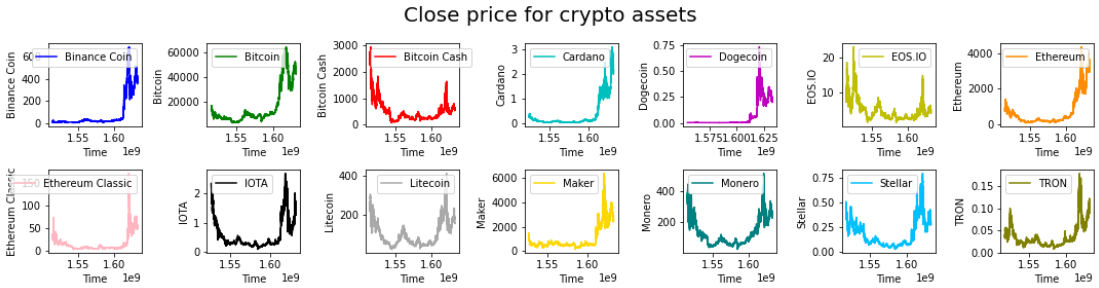
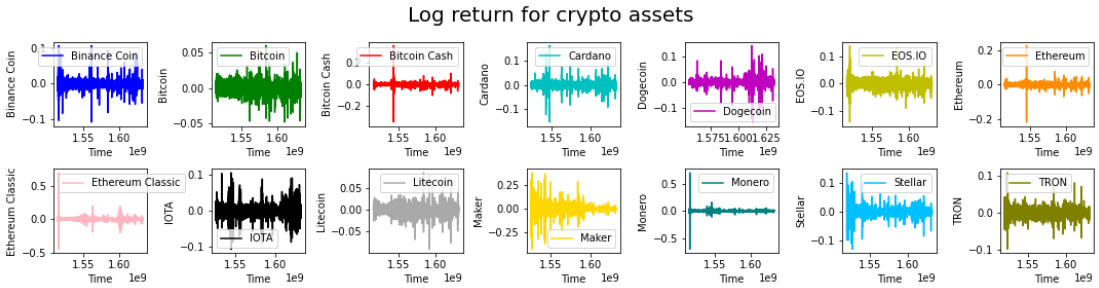


Figure 2 - Log Returns of 14 Cryptocurrencies Over Time



In terms of the log returns, the plots give intuitive observations of the stationary pattern of returns across different assets. However, a serious volatility clustering property is also observed, which means large changes in prices tend to cluster together, resulting in persistence of the amplitudes of price changes that can explain the huge price surge in Figure 1 to some extent.

We also explore the correlation of log returns between different cryptocurrencies on both train and test data to have deeper insight of asset dependency and diversification level of crypto market.

Interestingly, log returns on the train data show insignificant correlation in pairs of assets, which is in line with our expectation (Figure 3), while correlations on the test data are much higher. Therefore, we look more closely to rolling and shorter periods (for example, the test data), to know whether it is the time horizon or the specific period that leads to the difference. Figure 4 shows the correlation on a 6-month horizon, the co-movement of these cryptocurrencies in shorter term became more significant and reached a peak level in the most recent period (June-September 2021), aligning with the real-world attention that the cryptocurrency became more and more popular in the financial market especially start from year 2020.

Figure 3 - Asset Correlation on the Whole Period, Train & Test

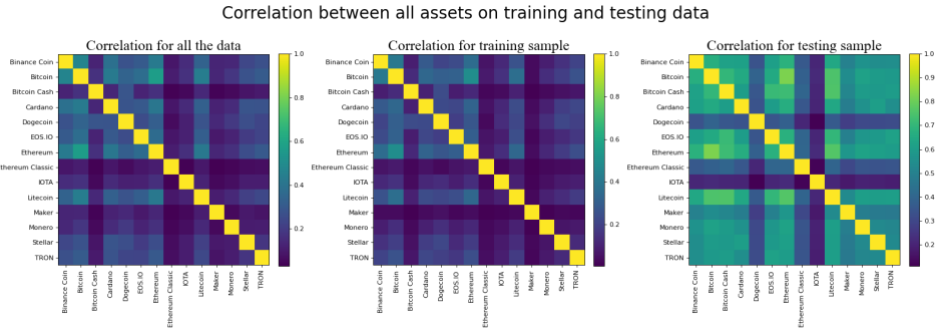
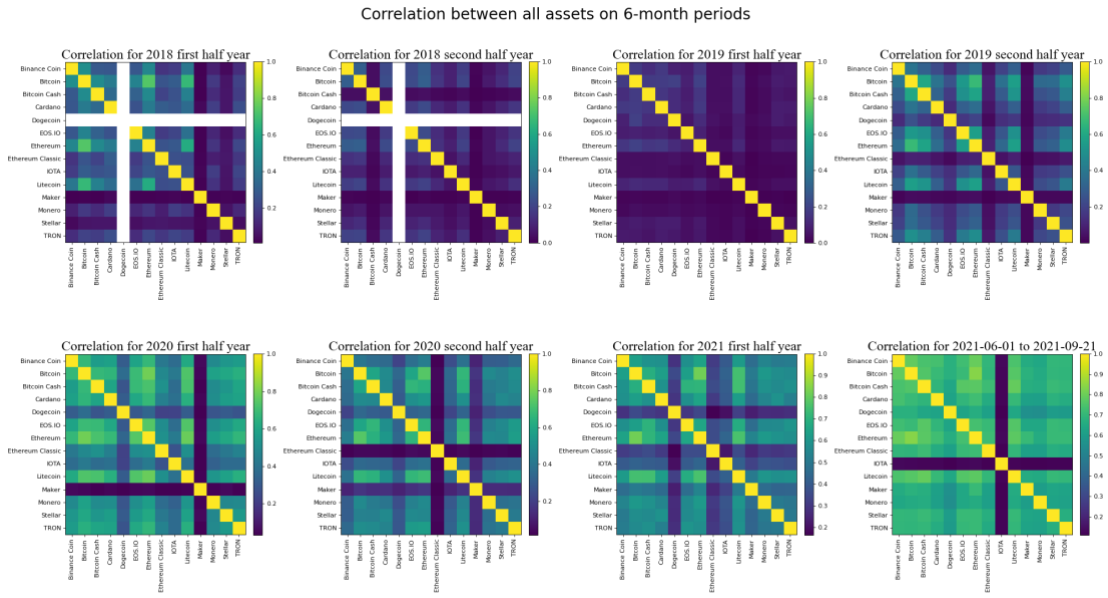


Figure 4 - Asset Correlation on 6-month Periods



4. Feature engineering

We construct 17 new futures based on the original 7 features in the training set, calculating methods and corresponding economic meanings are summarized in Table 2:

Table 2 - Feature Construction

Name	Construction Method	Economic Meaning
Day of Week	Mon-Sun: 0-6	whether data has day-of-the-week effects
weekend	Yes:1; No: 0	whether data has weekend effects
Month	Jan-Dec: 1-12	whether data has monthly effects
upper_Shadow	High - Max(Close, Open)	a tall upper shadow means a downturn is coming
lower_Shadow	Min(Close, Open) - Low	a tall lower shadow means a rise is coming
high_div_low	High/Low	fluctuation by min
trade	Close - Open	fluctuation by min
gtrade	trade/Count	min fluctuation relative to number of trades
shadow1	trade/Volume	min fluctuation relative to trading volume
shadow3	upper_Shadow/Volume	market sentiment factor
shadow5	lower_Shadow/Volume	market sentiment factor
log_return_1min	1 min log return	1-min performance
log_return_5min	5 min log return	5-min performance
log_return_1min_abs	absolute value of 1-min log return	1-min profit/loss
log_return_5min_abs	absolute value of 5-min log return	5-min profit/loss

After constructing all the features, we standardize all of them to facilitate the subsequent model training.

5. Model Building

The aim of this prediction is to predict the return of the price P^a for each asset a. For each timestamp in the dataset, we use the 'target' value for prediction: target. Which comes from log return R^a over 15 minutes (According to the definition provided by Kaggle contest).

$$R^a(t) = log\big(P^a(t+16)/P^a(t+1)\big)$$

if $M(t)$ is the weighted average market returns

$$M(t) = \frac{\sum_a w^a R^a(t)}{\sum_a w^a} \qquad \beta^a = \frac{\langle M \cdot R^a \rangle}{\langle M^2 \rangle}$$

the target is:

$$Target^a(t) = R^a(t) - \beta^a M(t)$$

(1) Baseline Models

a. Linear Model

Firstly, we try a simple linear regression model. Table 3 are the independent Pearson correlation result of all 14 assets respectively. The weighted Pearson correlation on test data is: **0.00865**.

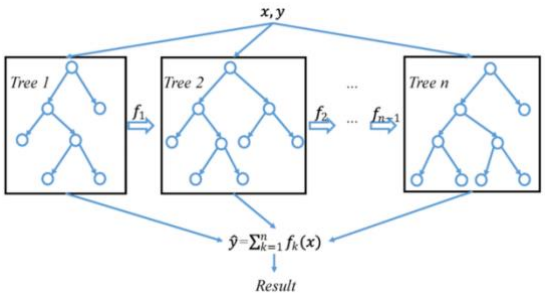
Table 3 - Test Correlation Scores of Linear Regression

Name	**Binance Coin**	**Bitcoin**	**Bitcoin Cash**	**Cardano**	**Dogecoin**	**EOS.IO**	**Ethereum**
Score	0.0081	0.0086	0.0336	0.0202	0.0023	-0.0020	0.0128
Name	**Ethereum Classic**	**IOTA**	**Litecoin**	**Maker**	**Monero**	**Stellar**	**TRON**
Score	-0.0220	0.0013	0.0210	-0.0165	0.0019	-0.0140	0.0198

b. XGBoost model

Next, we consider using XGBoost model, which can represent both first-order and second-order derivatives through second-order Taylor expansion. The regular term is added to control the complexity to avoid the overfitting problem. In addition, the strategy of random forest is also referred to in each iteration to support data sampling. The decision trees of random forest training are independent of each other, but XGBoost establishes a new tree by correcting the error of the previous decision tree.

Figure 5 - The Process of XGBoost Model



The model is as follows (J is the number of trees):

$$\widetilde{y_1} = m_j(\boldsymbol{x_i}) = \sum_{j=1}^{J} T_j(\boldsymbol{x_i})$$

The ultimate goal of XGBoost algorithm is to minimize the objective equation considering both the regular term and the error term:

$$0 = \sum_{i=1}^{I} loss(y_i, \widetilde{y_i}) + \sum_{j=1}^{J} \Omega(T_j)$$

Here are the results for each asset trained by XGBoost model and the weighted Pearson correlation on test data is: **0.0014129**.

Table 4 - Test Correlation Scores of XGBoost

Name	**Binance Coin**	**Bitcoin**	**Bitcoin Cash**	**Cardano**	**Dogecoin**	**EOS.IO**	**Ethereum**
Score	0.0083	-0.0055	-0.0110	0.0034	-0.0113	-0.0018	0.0109
Name	**Ethereum Classic**	**IOTA**	**Litecoin**	**Maker**	**Monero**	**Stellar**	**TRON**
Score	0.0150	0.0003	-0.0070	-0.0083	0.0042	0.0083	0.0111

c. Evaluation

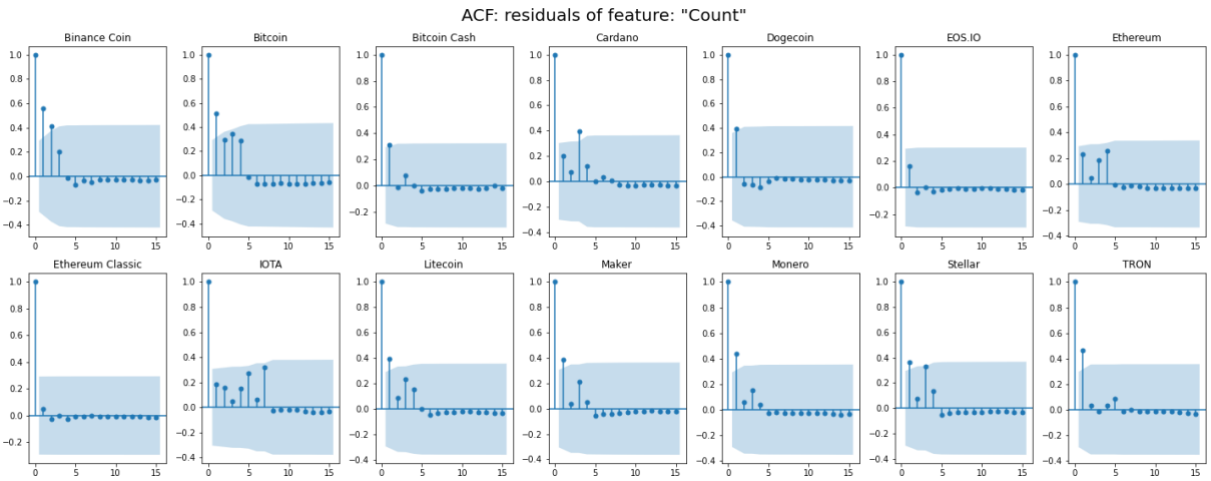
The results of the baseline models are not satisfactory. One possible reason behind is that the volatility clustering we observed in section *3. Exploratory data analysis* indicates a non-constant variance and non-normal distribution. Time-varying variances of features can lead to incomparability of scales after standardization (most features are super close to 0 and information contained shrinks). To support our conjecture, we perform normality and Engle's ARCH effect test to depicts whether and how variance changes with time. The Engle's test depends on whether the residuals of data have autocorrelation. If the residuals have autocorrelation, it means there exists ARCH effect.

We take the feature "count" as an example, resample the data by month, the test p-values and residual autocorrelation plots of the 14 assets are shown below:

Table 5 - Engle's Test and Normality Test Results - p-values

Asset Name	**Engle's**	**Normality**	**Asset Name**	**Engle's**	**Normality**
Binance Coin	0.000463	1.245278e-08	**Ethereum Classic**	0.683062	6.449586e-19
Bitcoin	0.00028	1.610225e-10	**IOTA**	0.001661	1.509755e-07
Bitcoin Cash	0.394082	1.386276e-11	**Litecoin**	0.000187	7.336070e-09
Cardano	0.000432	1.614855e-08	**Maker**	0.000632	5.531190e-09
Dogecoin	0.016426	1.905516e-09	**Monero**	0.000864	1.315559e-09
EOS.IO	0.996637	9.659301e-16	**Stellar**	0.00021	8.571107e-09
Ethereum	0.000213	6.349139e-09	**TRON**	0.159448	1.610225e-10

Figure 6 – ACF Plots



From the results, it is obvious that we should reject H_0 for most of the assets, which means the data is not normally distributed and volatility clustering exists. Hence, we refine our model by applying Box-Cox transformation.

(2) Model enhancement -- adding Box-Cox Transformation

a. Evaluation

In our work, Box-Cox transformation act as the method to transfer the non-normal independent variables into a normal shape, in order to obtain the constant variances for all features.

If *w* is our transformed variable and *y* is our target variable, then:

$$Y^* = h_\lambda(Y) = \begin{cases} \log(Y), \text{ if } \lambda = 0 \\ \dfrac{Y^\lambda - 1}{\lambda}, \text{ if } \lambda \neq 0 \end{cases}$$

After the transformation, the usual assumptions (normal and constant variances) for linear model are expected, that is:

$$h_\lambda(\boldsymbol{Y}) \sim N(\boldsymbol{XB}, \sigma^2 I)$$

b. Comparison

We can clearly see from Table 6 that Box-Cox transformation is effective and helps to improve the performance of our model significantly, from 0.008 to 0.01 for linear model and 3 times for XGBoost model.

Table 6 - Comparison of Linear Regression and XGBoost

	Linear Regression	**XGBoost**
No Box-Cox transformation	0.008651	0.001413
With Box-Cox transformation	**0.010551**	**0.004459**

6. Model extension – adding new features

In order to explore more possibilities, we added some new features to train our models. Here are the details of the new features:

Table 7 - New Features

Name	Construction Method	Economic Meaning
after	date after 2018/6/4	Response after Elon Musk tweeted about crypto
afterdoge	date after 2021/3/29	Response after Elon Musk tweeted about dogecoin
shadow2	upper_Shadow /Low	downward trend factor
shadow4	lower_Shadow/High	upward trend factor
Closing_moving_average	lower_shadow/Volume	5-day moving average price

Disappointingly, new added features do not give better results, indicating a potential overfitting problem. We still choose linear regression with originally constructed feature and Box-Cox transformation as our final model.

Table 8 – Comparison of Models with New Features

	Linear Regression	XGBoost
Originally constructed features	**0.010551**	**0.004459**
More features	0.008894	0.004127

7. Future Work

With no further improvement in the extension model with new features, and the final model also do not give promising result, we should consider dealing with potential collinearity problems. Two methods can be adopted: LASSO and ridge regression for feature selection and regularization.

8. Reflection and Conclusion

The final model of this project is the linear regression model with originally constructed feature and Box-Cox transformation which gives 0.010551 of weighted correlation. The main problem of this project has 2 characteristics: phenomenon of volatility clustering and high-frequency data. These 2 properties make it far more difficult to capture the crucial features over time as well as in a tiny time interval than usual machine learning tasks. Furthermore, methods of feature selection and regularization can be further adopted to eliminate the collinearity between features.

References

1. https://www.kaggle.com/lucasmorin/on-line-feature-engineering/notebook

2. https://www.kaggle.com/cstein06/tutorial-to-the-g-research-crypto-competition

3. https://www.kaggle.com/yamqwe/g-research-xgboost-starter-notebook?scriptVersionId=81124879

4. https://www.kaggle.com/swaralipibose/new-features-training-notebook-and-feature-eval

5. https://www.kaggle.com/swaralipibose/new-features-eda-using-elon-musk-and-crypto-trends#Load-Data

Contribution:

Coding:

Data processing, model training&evaluation: WANG Zhongchen

Time series analysis and hypothesis testing: WANG Lei

Report writing:

YE Xiaoyu & ZHANG Quandi