Project 3: G-Research Kaggle Contest

LIU Chen
20809024
cliudh@connect.ust.hk

1 Introduction

This report contains experiments and analysis on the Kaggle Context held by G-Research(`https://www.kaggle.com/c/g-research-crypto-forecasting/overview`). This contest offers the one-minute kline information for several cryptocurrency spots or related derivatives. The goal is to use these information to build a model that can predict the future trend of the assets. In detail, the kline information includes the price at the beginning of the minute, the highest price during the minute, the lowest price during the minute, the price at the end of the minute. Meanwhile, the dataset also contains the number of cryptoasset units traded during the minute as well as the volume weighted average price for the minute. The prediction target is the 15 minute residualized returns.

In a more formal way, given raw input X and target Y, we need to get a function $g(\cdot)$ that transforms X into more useful feature $g(x)$ and prediction model $\phi(\cdot)$ that the prediction $\phi(g(x))$ can be used to judge the future tendency of Y. So for this contest, the core lies in the construction of input features as well as the tunning of model. In this report, the discussion of the two factors will be present. In specific, this report puts emphasis on several questions. For the first part, how to construct good factors is very important in Quantative Finance. While most of them are designed for some classical assets such as stocks. A natural question is whether the factors can still have power in predicting crypto assets. In addition, some analysis of the model is also a must. As we all know, when the train and test environments are similar and little domain gap exists, larger and more complicated model with suitable training skills leads to better performance. However, for varying environment, it can lead to a different pattern. In this report, analysis of this issue is also given. The last but not the least feature selection is widely used in machine learning to find informative subset of feature. In this report, the efficacy of feature selection is verified on the scenario of predicting the profit of crypto assets. In detail, the point is whether feature selection can enhance the model's performance.

To sum up, this report contains the discussion of the following parts.

- whether the classical factors have predictive ability on crypto assets?
- whether the increase of model capacity leads to better strategy ?
- whether the can we use the feature selection to get a better model?

2 Dataset Description

The given data contains 14 kinds of assets such as Bitcoin Cash, and Ethereum. For better illustration, Figure. 1 gives several examples of the data. In detail, **timestamp** stands for the time in millisecond which is widely used in UNIX system. the **Asset ID** denotes the corresponding id of asset such as 0 for Binance Coin. **Count** stands for the number of trades in that minute. **Open** and **Close** mean the price at the beginning and the end. **High** and **Low** mean the highest and lowest price inner the minute. **Volume** means the The number of cryptoasset units traded during the minute. **VWAP** means the volume weighted average price for the minute. And finally, **Target** is the goal we want to predict. The Target is the 15 minute residualized returns. In specific, it firstly calculates the log return for one

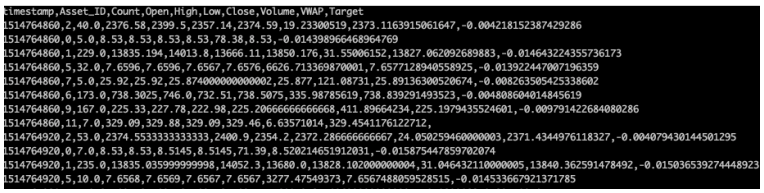

Figure 1: This figure shows the examples of data from the G-Research Kaggle Contest.

asset a as follows,

$$R^a(t) = log(\frac{P^a(t+16)}{P^a(t+1)}) \tag{1}$$

And to get a residue, the holder suggests that the target should drop the effect of other assets. Given some weights of different assets, a weighted average can be calculated as follows,

$$M(t) = \frac{\sum w^a R^a(t)}{\sum w^a} \tag{2}$$

And the final target is calculated as $R^a(t) - M(t)\beta^a$. Here β^a is calculated via rolling.

3 Basic Setting

In this report, both results and analysis will be illustrated. Due to the restriction of submission trials, a large proportion of results are shown in local validation. And local validation use the correlation with ground truth. For the model of each kind with best validation result, the submission performance will be reported.

Note that due to the leakage of label as mentioned in the discussion(`https://www.kaggle.com/julian3833/proposal-for-a-meaningful-lb-strict-lgbm?scriptVersionId=80421622`). To get more meaningful score, the data partition should be altered. In detail, only the data before **2021-06-13 00:00:00** are kept for train and validation. For local validation, a subset of validation is used. In this part process, the training data is the first 75% part and remaining part is used for testing. This part of validation is used for find good model and suitable parameters. And note that this problem is formalized as a regression problem.

4 Results and Analysis

4.1 Experiment Setting

In this report, two methods are selected. One is Linear Regression, this report utilizes the implementation of Scikit-Learn [2]. The other one is Lightgbm [1]. For this contest's data, there exists some null values, in this report the rows with null values are dropped. For this contest, all the assets should be predicted, this report use different models for different assets. But the model types are the same. In this report correlation between prediction and target is reported.

4.2 Analysis on Some Classical Feature

To begin with, raw features about the prices are fed into Linear Regression Model directly, the validation results are shown in Table. 1. Here raw feature means using the price information in that minute without any modifications. For better illustration, only 5 of the assets are shown in the table. Here set1 adds some classical features. In detail, two features are added to evaluate the spread of the price. The first one is the Price high minus the the maximum of Price open and close,

$$Feature_1 = P_{High} - max(P_{Open}, P_{Close}) \tag{3}$$

Similarly, the other one is,

$$Feature_2 = min(P_{Open}, P_{Close}) - P_{Low} \tag{4}$$

The results are reported in Table. 1 And set2 adds some new features from (`https://www.kaggle.com/kartik2khandelwal/feature-engineering-improved-score`). In detail, the spread between the end and the beginning is selected,

$$Feature_3 = P_{Close} - P_{Open} \tag{5}$$

$$Feature_4 = \frac{P_{Close}}{P_{Open}} \tag{6}$$

In addition, the spread between the high and the low is also selected,

$$Feature_5 = P_{High} - P_{Low} \tag{7}$$

$$Feature_6 = \frac{P_{High}}{P_{Low}} \tag{8}$$

It can be observed that using the features of set2 can lead to a more significant enhancement,

Feature	Binance Coin	Bitcoin	Bitcoin Cash	Ethereum	Dogecoin
Raw	0.0048	0.0007	0.0148	0.0175	0.0073
Raw+Set1	0.0081	-0.0009	0.0122	0.0174	0.0039
Raw+Set2	0.0063	-0.0046	0.0185	0.0127	0.0156
Raw+Set1+Set2	0.0065	-0.0061	0.0152	0.0123	0.0138

Table 1: This table shows the correlation of using different features with linear regression.

especially in Dogecoin. The combination of Set1 and Set2 does not add to some enhancement, and here Raw+Set2 is selected. The reason why Set2 can work is that the ratio or the residue offers a measurement of the volatlity in that minute which adds to the information. And the answer to the first question is that some classical factors do work.

4.3 Analysis on Model Capacity

Using the best feature set, i.e. Set2, given in the previous setion, this section compares simple model, i.e. Linear Regression and relatively complicated model, i.e. Lightgbm. For Lightgbm, three hyperparameters determine the cpacity, i.e. number of leaves(nl), max depth(md) and number of estimators(ne). In this part, the max depth is controlled as 3. And number of leaves selects values from $\{10, 50, 100\}$. The number of estimators selected values from $\{10, 50, 100\}$. The results are shown in Table. 2. It can be found that using Lightgbm does enhance the performance on some assets such as Bitcoin. But some cases also show the phenomenon of degradation. It may be related with the complexity of data. For Bitcoin, many traders put emphasis on it due to the value and capacity. And the competition reduce the power of simple models. For some other assets such as Dogecoin, Linear Regression can get a decent results. Furthermore, increasing the capacity via increasing the number of estimators have few enhancement in the validation set. It is in accordance with intuition that the trading environment is varying, overfiting is not a good choice here.

Feature	Binance Coin	Bitcoin	Bitcoin Cash	Ethereum	Dogecoin
Linear Regression	0.0063	-0.0046	0.0185	0.0127	0.0156
Lightgbm nl=10,ne=10	0.0165	0.0054	0.0018	0.0073	0.0092
Lightgbm nl=10,ne=50	0.0105	0.0134	0.0082	0.0104	0.0057
Lightgbm nl=10,ne=100	0.0068	0.0112	0.0098	0.0092	0.0068
Lightgbm nl=50,ne=10	0.0165	0.0055	0.0018	0.0073	0.0092
Lightgbm nl=50,ne=50	0.0105	0.0134	0.0082	0.0104	0.0058
Lightgbm nl=50,ne=100	0.0067	0.0112	0.0099	0.0092	0.0068
Lightgbm nl=100,ne=10	0.0165	0.0055	0.0018	0.0073	0.0092
Lightgbm nl=100,ne=50	0.0105	0.0134	0.0082	0.0104	0.0058
Lightgbm nl=100,ne=100	0.0067	0.0113	0.0099	0.0092	0.0068

Table 2: This table shows the correlation of using different methods with Raw+Set2 feature.

4.4 Analysis on Feature Selection

For the feature selection of Linear Regression, Lasso [3] is used to find For Lasso, it adds a L_1 regularization to the loss function in the form of $\|y - \beta X\|_2^2 + \alpha\|\beta\|_1$. important feature. The implementation of Scikit-Learn is seleted in this report. For the feature selection of Lightgbm, the selection is according to the feature importance given by the algorithm. In Table. 3, the results show that using feature selection leads to better performance. In such kind of data, it can be very noisy, using Lasso in a right way leads to the performance boost. And the Lasso with $\alpha = 0.01$ is the final model for linear regression.

Feature	Binance Coin	Bitcoin	Bitcoin Cash	Ethereum	Dogecoin
Linear Regression	0.0063	-0.0046	0.0185	0.0127	0.0156
Lasso $\alpha = 0.005$	0.0293	-0.0206	0.0146	0.0244	0.0567
Lasso $\alpha = 0.01$	0.0292	-0.0225	0.0146	0.0244	0.0568
Lasso $\alpha = 0.05$	0.0292	0	0	0.0244	0.0578
Lasso $\alpha = 0.1$	0.029	0	0	0.0244	0.0588

Table 3: This table shows the feature selection result of linear regression.

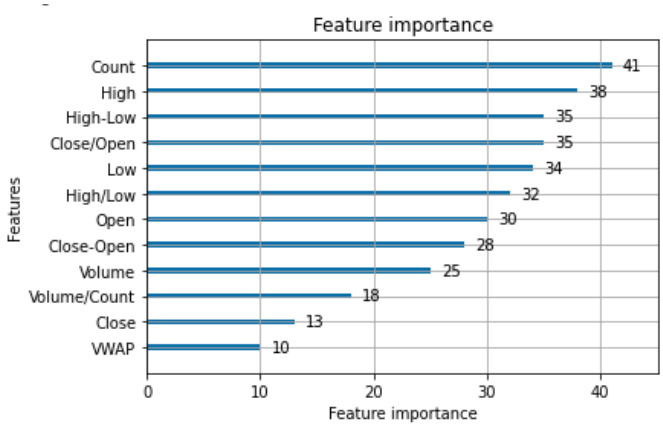

Figure 2: This figure shows the feature importance of Bitcoin.

For Lightgbm, the one with number of leaves 100 and number of estimator 50 is chosen. The feature importance is shown in Figure. 2 and Figure. 3. For clarity, here only show the figure of Bitcoin and Dogecoin. It can be observed that Count, High-Low,High/Low, Low and High keep competitive for different assets. Here these 5 factors are picked to get a new model. The results are shown in Table. 4. For Lightgbm, feature selection does not give a boost. It may need some alternation.

Feature	Binance Coin	Bitcoin	Bitcoin Cash	Ethereum	Dogecoin
Full	0.0105	0.0134	0.0082	0.0104	0.0058
Select Feature	0.0006	0.0156	0.0010	-0.004	0.0087

Table 4: This table shows the correlation of using different features with Lightgbm.

4.5 Final Score

At the end, the best model of Linear Regression and Lightgbm is submitted to the contest. And the scores are shown in Figure. 4. It can be observed that Lightgbm is not stronger than Linear Regression.

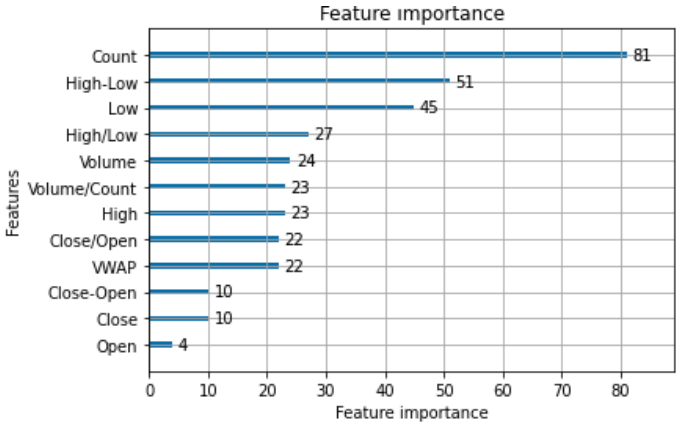

Figure 3: This figure shows the feature importance of Dogecoin.

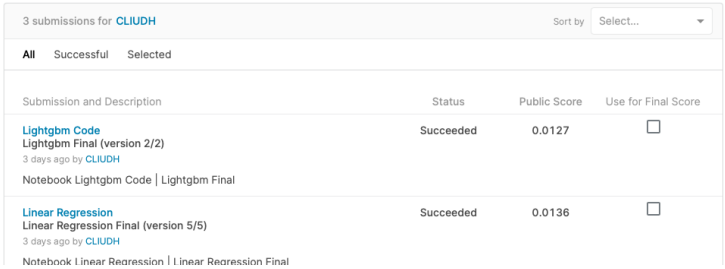

Figure 4: This figure shows the submission results of two final models.

5 Conclusion

As stated at the start, this report is concerned about four issues, and the previous can give naive answers for them. For the first question, some classical features still have prediction power for crypto assets. For the second question, this dataset is very likely to be overiftted, so increasing the capacity to much can lead to performance degradation. For the third question, feature selection can be quite useful for Linear Regression, it may be caused by colinearity of features or some noise from the design of feature. For the experiment in this report, feature selection for Lightgbm does not give a boost.

References

[1] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30:3146–3154, 2017.

[2] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

[3] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.