G-Research Crypto Forecasting

Ricky Chan Tsz Wai

Department of Mathematics

Introduction

The rapid evolution of financial markets, especially in the domain of cryptocurrencies, presents both challenges and opportunities for predictive analytics. The volatile and low-Signal-to-noise-ratio nature of these markets demands sophisticated modeling techniques that can capture complex patterns and provide reliable forecasts. This research aims to **refine cryptocurrency return predictions** by leveraging a robust methodology that spans from traditional regression models to cutting-edge ensemble machine learning methods and deep learning methods.

Feature Engineering

Feature engineering plays a pivotal role in enhancing the performance of predictive models in the financial markets. By creatively transforming and expanding the original data set through feature engineering, analysts and traders can capture **more complex patterns and relationships that affect asset prices.** This process often involves deriving new variables from existing data, such as moving averages, momentum indicators, and interaction effects between different market variables, which can significantly improve the predictive accuracy and robustness of trading models.

However, in this project, relying solely on **Open, High, Low, and Close (OHLC)** data in financial modeling can be **limiting**. OHLC data primarily captures the price movements within specific time frames but often misses underlying market dynamics. It does not account for the volume of trades or the **volatility** during the trading period, which can provide critical insights into the strength of price movements.

$$\text{return\_}l\text{m} = \left(\left(\frac{\text{Close}_t}{\text{Close}_{t-l}} - 1\right)_{\text{clip}(q_{0.01}, q_{0.99})} + 1\right)^{\frac{1}{l}} - 1$$

$$\text{True Range} = \max((\text{High} - \text{Low}), |\text{High} - \text{Close}_{\text{previous}}|, |\text{Low} - \text{Close}_{\text{previous}}|)$$

$$\text{ATR} = \text{SMA}_{14}(\text{True Range})$$

$$\text{EMA}_t = (\text{P}_t \times \alpha) + (\text{EMA}_{t\text{-}1} \times (1 - \alpha))$$

$$\text{RS} = \frac{\text{Average Gain over N periods}}{\text{Average Loss over N periods}}$$

$$\text{MACD} = \text{EMA}_{12}(\text{Close}) - \text{EMA}_{26}(\text{Close})$$

$$\text{Signal Line} = \text{EMA}_9(\text{MACD})$$

$$\text{RSI} = 100 - \left(\frac{100}{1 + \text{RS}}\right)$$

$$\text{BBBand Up} = \text{SMA}_{20}(\text{Close}) + (2 \times \text{SD}_{20}(\text{Close}))$$

$$\text{BBBand Mid} = \text{SMA}_{20}(\text{Close})$$

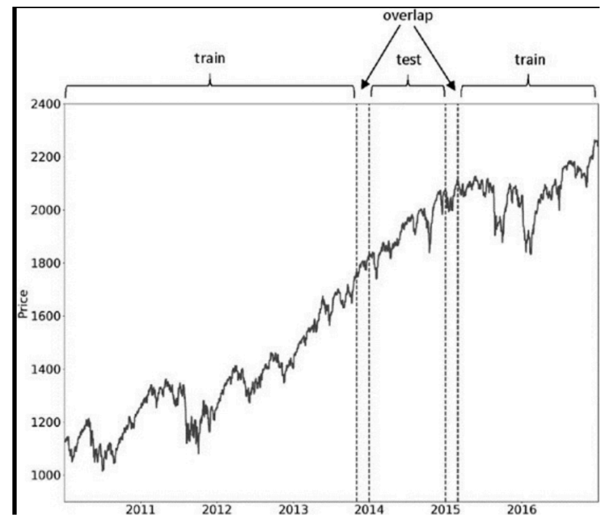
$$\text{BBBand Low} = \text{SMA}_{20}(\text{Close}) - (2 \times \text{SD}_{20}(\text{Close}))$$

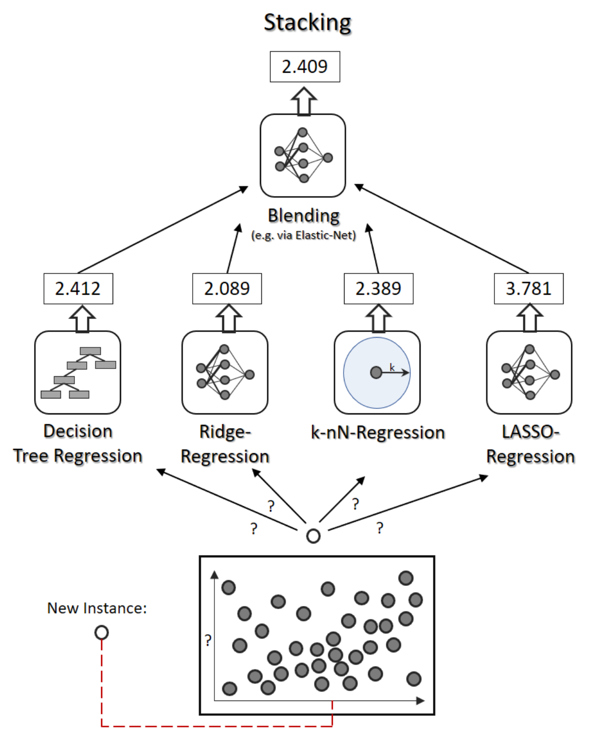
Therefore, we try our best to engineered the OHLC data into some technical indicators.

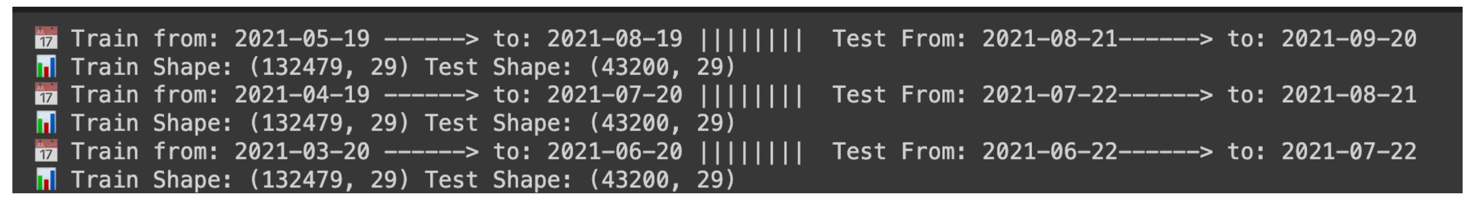
Methodology and Design

Forecasting future returns is a challenging task, especially in financial market due to its stochastic feature. However, when designing such learning task where validation involved, data leakage/using future data is a common mistake that people often makes. Therefore we design a **cross validation method specialized for time series**.

"The purpose of cross-validation (CV) is to determine the generalization error of an ML algorithm, so as to prevent overfitting. CV is yet another instance where standard ML techniques fail when applied to financial problems. Overfitting will take place, and CV will not be able to detect it. In fact, CV will contribute to overfitting through hyper-parameter tuning.







```
📅 Train from: 2021-05-19 ------> to: 2021-08-19 ||||||||  Test From: 2021-08-21------> to: 2021-09-20
📊 Train Shape: (132479, 29) Test Shape: (43200, 29)
📅 Train from: 2021-04-19 ------> to: 2021-07-20 ||||||||  Test From: 2021-07-22------> to: 2021-08-21
📊 Train Shape: (132479, 29) Test Shape: (43200, 29)
📅 Train from: 2021-03-20 ------> to: 2021-06-20 ||||||||  Test From: 2021-06-22------> to: 2021-07-22
📊 Train Shape: (132479, 29) Test Shape: (43200, 29)
```

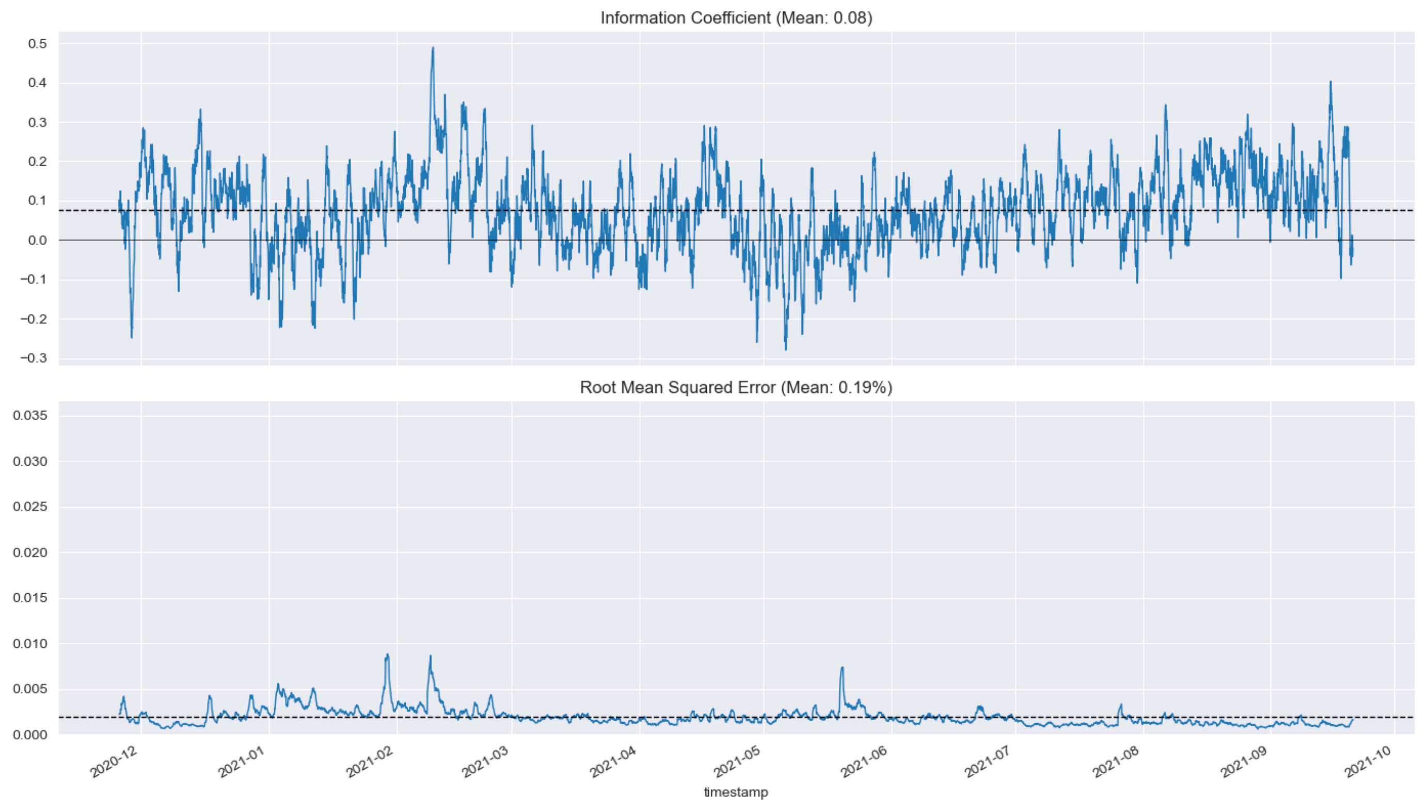
First, we selected few simple linear models for baseline studying, including **linear regression, ridge regression and lasso regression**. Then, we test those model on raw data then on the engineered data.

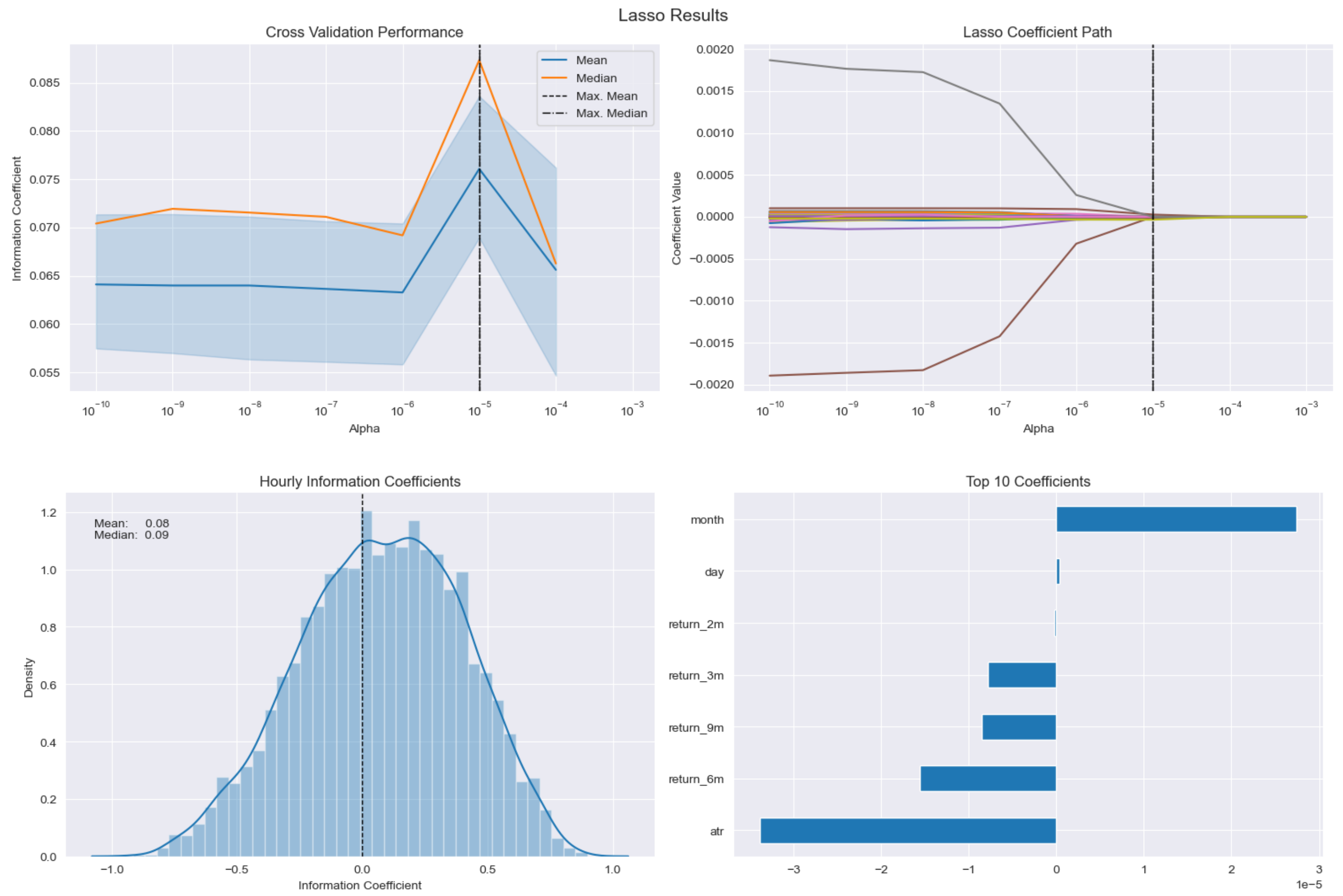
Second, we choose few **boosting models** (catboost, Xgboost, ExtraTree,lightgbm) to check if they could learn extra information. At last we **stack** them into a regressor so as to improve the model performance.

Lastly, we tried deep neural nets. Again we first use **fully connected network** as the baseline, then we design a **lstm structure** to model the data. We first **transform the data into 'sentence' and label them.** Then we used **warmup technique** to control the learning rate for better leanrning.

RESULTS

The Information Coefficient (IC) is a crucial metric in finance used to measure the strength and direction of the predicted relationship between forecasted and actual asset returns. It is essentially a correlation coefficient that ranges from -1 to 1, where **a higher absolute value indicates a stronger predictive power of the model**. A positive IC suggests that forecasts are directionally aligned with actual outcomes, improving the chances of making profitable investment decisions, while a negative IC implies inverse predictions.





CONCLUSION

LASSO outperforms all other models including deep learning one. I would say the features itself already contained lots of noise, adding more layers/making the model more complex would probably fit the noise seriously resulting in **bad predicting power**(low ic), but 'good' metric (low rise).

Deeper isn't always the best choice for learning, especially in financial market. By my experience in the industry, **feature engineering/factor mining** is the key of reducing noise. Making the data with **higher frequency** like in nano seconds would achieve that goal also.

Model	Information Coefficient	RMSE
LASSO - FE	0.08	0.19%
Ridge - FE	0.07	0.19%
CatBoost - FE	0.07	0.24%
LGBM - FE	0.07	0.24%
Linear Regression - FE	0.0624	0.2%
XGBoost - FE	0.06	0.30%
Ensemble - FE	0.05	0.19%
Linear Regression - RAW	0.04	0.19%
LASSO - RAW	0.03	0.19%
Ridge - RAW	0.03	0.19%
DNN - FE	0.01	0.37%
LSTM - FE	0.01	0.19%

REFERENCE

"Advance in Financial Machine Learning"

FUTURE WORK

1. Use **L2 data** (snapshot of order book, spread, imbalance..)
2. **AutoFE** for feature engineering
3. AutoML

Contribution

Coding: Ricky Chan Tsz Wai

Modeling: Ricky Chan Tsz Wai

Data Engineering: Ricky Chan Tsz Wai