Cryptocurrency High-Frequency Trading Strategy based on Orderbook Behavior

Lynn, 朱致伶

2024/10/07 ML_FinTech Project Slide

Outline

1. Motivation
2. Design of Analysis
3. Data
    - Pre-processing
    - Feature engineering
    - Exploratory data analysis
4. Method
5. Experiment and Result
6. Strategy Design and Back Testing
7. Conclusion

Motivation

- Cryptocurrency market is renowned for its high volatility and fragmented liquidity across different exchanges.
- Orderbook Behavior which reflects real-time market, related to price, volume and liquidity conditions.
- Goal: Develop a **cryptocurrency trading predictive model with** orderbook data.

- Why the project is important:
  - Study on Orderbook data and behavior
  - Risk Management through Liquidity
  - Practical use of machine learning model

Design of Analysis

Data Collection and Preprocessing

Crawl order book data From Binance API, implement rolling window approach.

Data Featuring

Design and craft meaningful feature from raw data.

Model Training and Optimization

Design and train different models, including:

- Linear regression
- Super vector machine
- XGBoost
- Long-short term machine

Model Evaluation

Assess model performance employed metrics **MSE, RMSE, MAE, R^2.**

Strategy Design and Back Testing

Design trading strategy based on trained model and back test model with market indicators.

- Sharpe Ration (SR)
- Maximum Drawdown (MDD)
- Profitability (Win Rate)

Conclusion

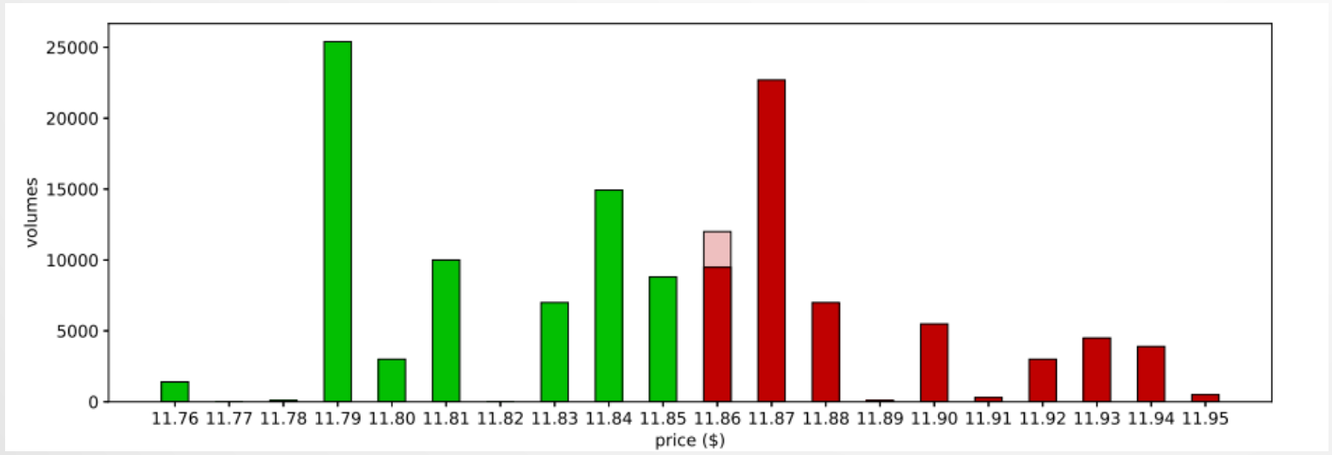
Interpret model performance and propose a strategy.

Data

- [Binance Order Book API](#) : BTC/USDT
- Response example

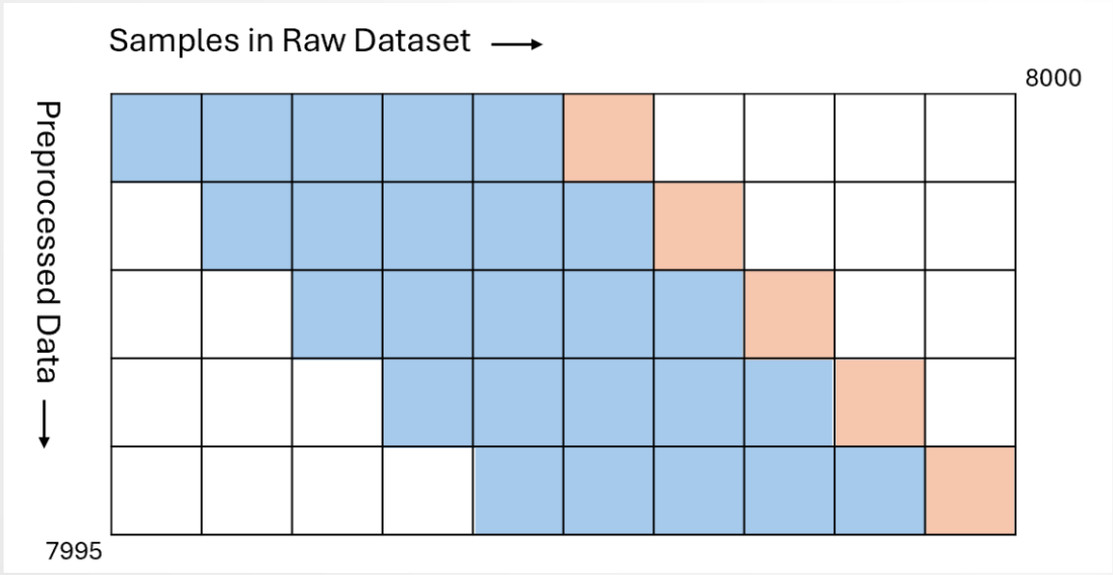
```
{
  "T": 1589436922972,   // transaction time
  "u": 37461            // update id
  "bids": [             // Buy order
    [
      "1000",           // Price
      "0.9"             // Quantity
    ]
  ],
  "asks": [             // Sell order
    [
      "1100",           // Price
      "0.1"             // Quantity
    ]
  ]
}
```

- Order book data schematic diagram



Data and Pre-processing

- Bitcoin (BTC) paired with Tether (USDT)
  - Market Dominance
  - Liquidity
  - Price Stability
  - Data Quality
- Implement rolling window approach

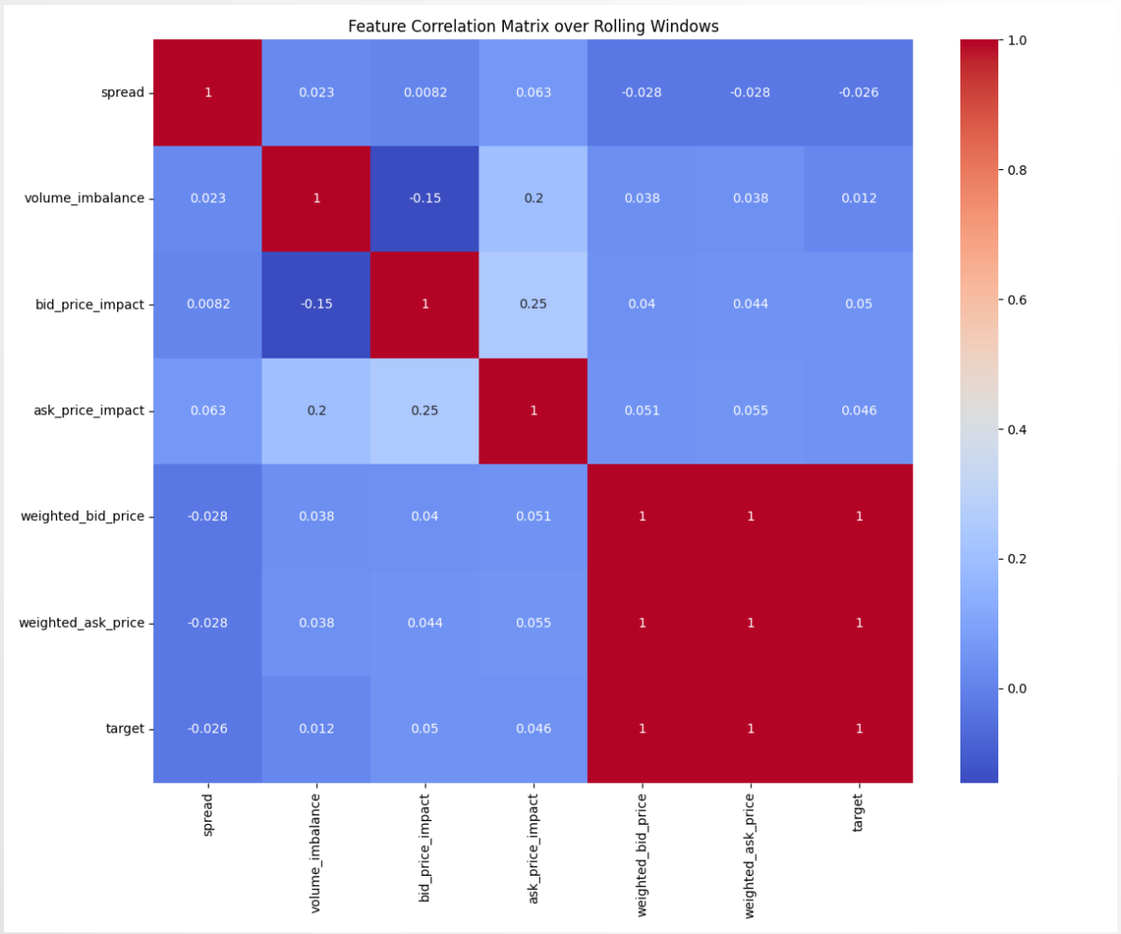


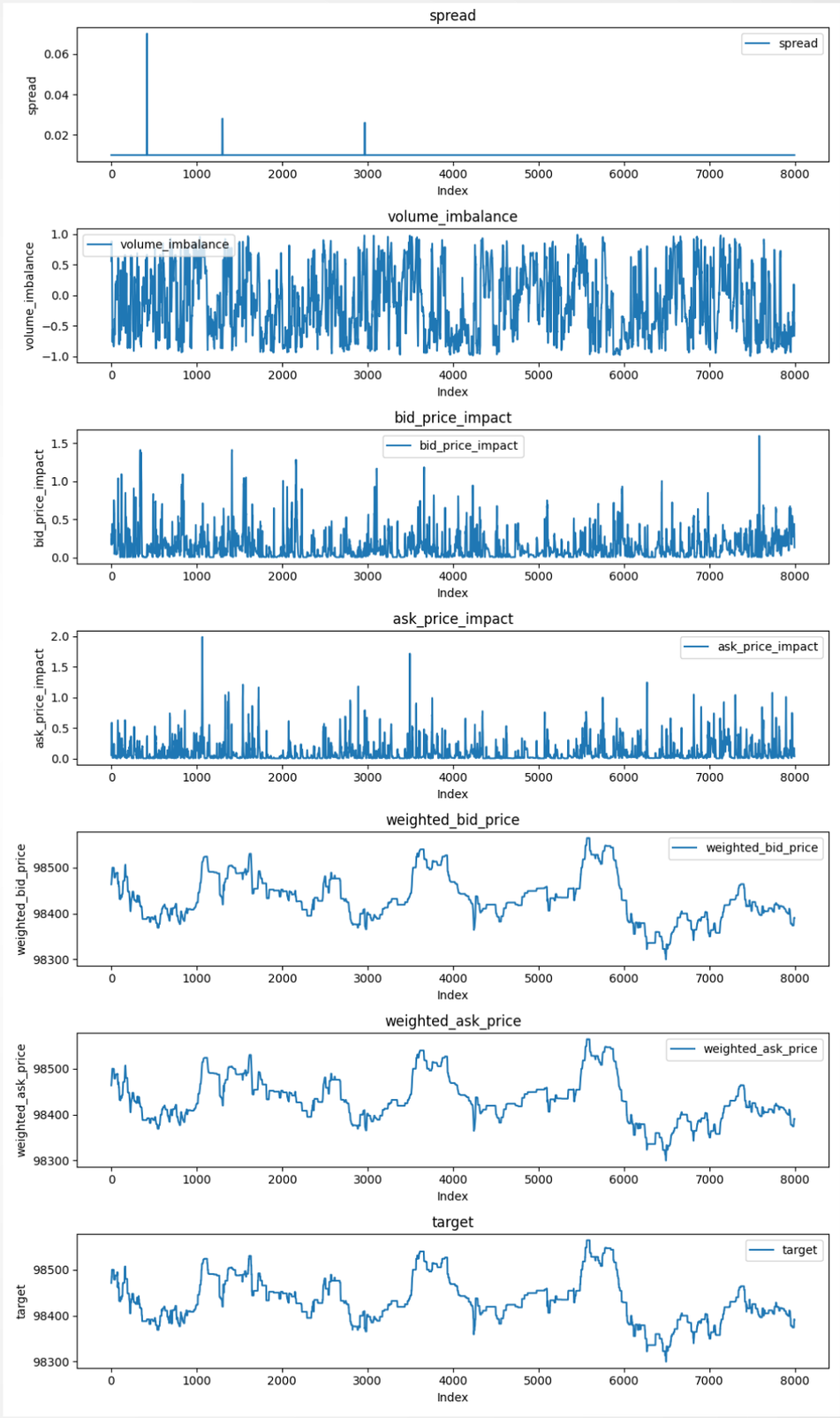
Feature Engineering

- Extract several features from the raw order book dataset for better model learning performance.
- To explicitly show cryptocurrency micro-market structure:
  - $Mid\ Price(Target) = {}^{P_{ask}+P_{bid}}/_{2}$
  - $Spread = P_{ask} - P_{bid}$: Product value difference between buyer and seller
  - $Volume\ Imbalance = \frac{\sum Vol_{ask} - \sum Vol_{bid}}{\sum Vol_{bid} + \sum Vol_{ask}}$: Strength of volume market trend
  - $Price\ Impact = \frac{\sum(V_i \times |P_i - P_{mid}|)}{\sum V_i}$
  - $Volume\ Weighted\ Price = \frac{P_i \times V_i}{\sum V_i}$

Exploratory Data Analysis

- Data distribution
- Correlation heatmap





Method

- Split dataset to training(70%) and testing(30%)
- Train and optimize the below machine learning model

Method	Purpose	Advantages	Disadvantages	Use Cases
Linear Regression	Predict continuous values	Simple, **interpretable**, efficient	Assumes linearity, sensitive to outliers	Price prediction, trend analysis
Support Vector Regression (SVR)	Predict continuous values	Effective in high dimensions, **flexible kernel**	Computationally expensive, sensitive to parameters	Stock price prediction, regression tasks
XGBoost	Boosting for predictions	Handles **large datasets**, regularization	Complex to tune, overfitting possible	Classification, regression tasks
LSTM	Sequence prediction	Captures **temporal dependencies**	Computationally expensive, requires tuning	Time series, speech recognition

Method

Linear regression

- Predicts a continuous target variable by fitting a linear equation to the data.
- Formula:

$$\boldsymbol{y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n + \epsilon}$$

where y is target, x's are the features, β_0 is the intercept, β_i are the coefficients.

Super-vector Regression

- Predicts a continuous target variable while minimizing the margin of error, using a margin of tolerance.
- Formula:

$$\boldsymbol{y = \sum_{i=1}^{n} (\alpha_i - \alpha_i^*) K(x_i, x) + b}$$

where y is target, $K(x_i, x$)is the kernel function, and α_i , α_i^* are Lagrange multipliers.

Method

- XGBoost(Extreme Gradient Boosting):
  - A decision-tree-based ensemble machine learning model optimized with gradient boosting.
  - Formula (Objective):

$$Obj = \sum_{i=1}^{n} L(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k)$$

where L is the loss function, $\hat{y}_i$ is the predicted value, and $\Omega(f_k)$ is the regularization term for tree f_k.

- Long Short-Term Memory (LSTM):
  - A type of recurrent neural network (RNN) used for sequence prediction tasks.
  - Formula (Cell State):

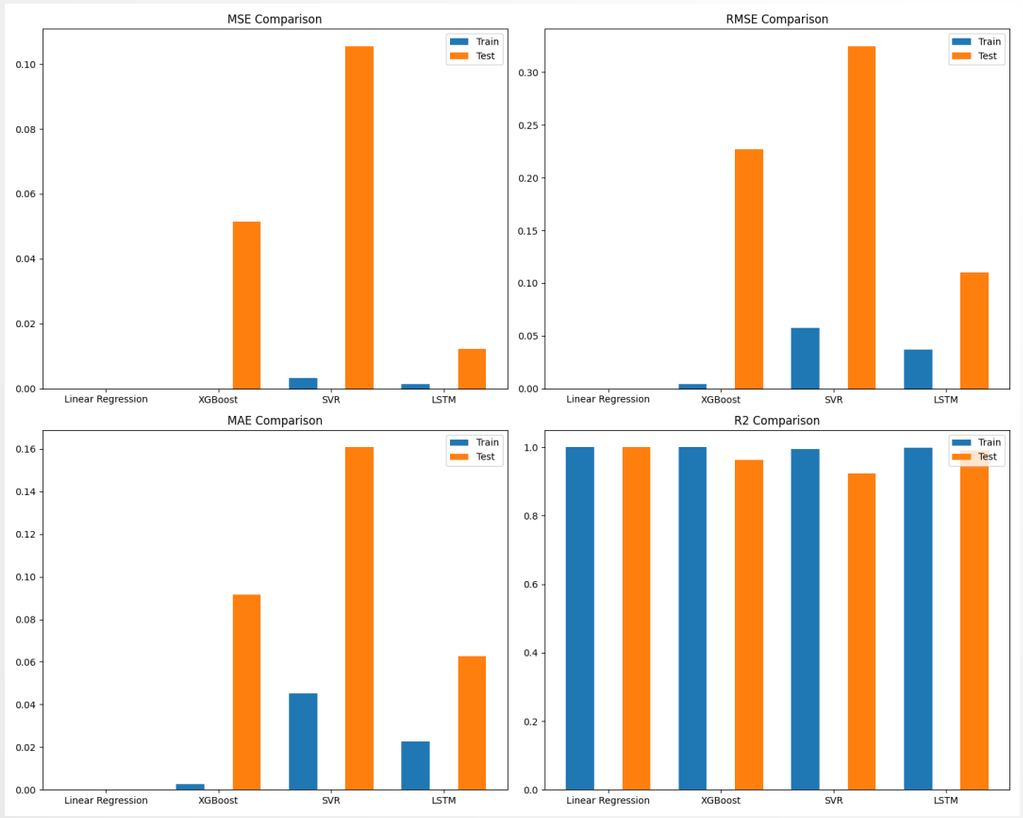
$$\boldsymbol{c_t = f_t \odot c_{t\text{-}1} + i_t \odot \tilde{c}_t}$$

where $f_t, \mathrm{i_t}, c_t$ are the forget, input, and candidate cell states, and $\odot$ represents element-wise multiplication.

Model Performance Evaluation

Model Performance on Testing Dataset

	MSE	RMSE	MAE	R2
LR	0.0	0.0	0.0	1.0
XGBoost	0.0	0.017	0.01	0.999
SVR	0.015	0.121	0.063	0.952
LSTM	0.002	0.042	0.028	0.994



MSE: Mean Square Error
RMSE: Root Mean Square Error
MAE: Mean Absolute Error
R2: R square

Trading Strategy and Back Testing

- Predictions from the models are translated into trading signals based on thresholds (θ) derived from historical analysis with **θ = 0.001**:
- Trading strategy determine the **best time** to buy/sell cryptocurrency with **Price Rate of Change per second**.
  - **Signal = 1** if (P_pred - P_current) / P_current > θ
  - **Signal = -1** if (P_pred - P_current) / P_current < -θ
  - **Signal = 0** otherwise
- A **stop-loss** mechanism is integrated to mitigate potential risks from adverse price movements.

Back Testing

- Market indicators to implement back testing. The strategy parameters, including θ, are optimized during back testing.:
- **Sharpe Ratio (SR)**: Measures the **risk-adjusted return** of an investment. A higher Sharpe ratio indicates better risk-adjusted performance.

$$SR = \frac{R_p - R_f}{\sigma_p}$$

where R_p is the portfolio return, R_f is the risk-free rate, and σ_p is the standard deviation of the portfolio's excess return.

- **Maximum Drawdown (MDD)**: Represents the maximum observed loss from a peak to a trough before a new peak is reached.

$$MDD = \frac{Peak\ Value\ - Trough\ Value}{Peak\ Value}$$

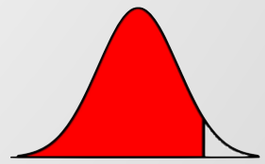
Back Testing and Mocked Live Trading Implementation

- **Profitability (Win Rate)**: Measures the percentage of trades that are profitable.

$$Profitability = \frac{Number\ of\ Winning\ Trade}{Total\ Number\ of\ Trade} \times 100$$

- Range of Market Indicators (XGBoost in real-world live trading):

Metric	Common Range	Good Trading Strategy (Good/Excellent)	XGBoost
Sharpe Ratio	0 ~ 3	>1.5 / >2.0	-0.034
Maximum Drawdown	10% ~ 100%	>20% / >10%	80%
Profitability	40% ~ 60%	>55%	-8%



Conclusion

In the project, …

- Crawl meaningful orderbook feature from raw data.
- Train four machine learning model, and compare the performance. XGBoost has the best model performance among four ML model.
- Develop trading strategy based on learned model.
- Implement back testing on trained XGBoost in real-world live trading environment, model performance is under expectation.

Future Work

- Study relationship between simple and complex data and learning model.
- Design and modify ML structure, tune model parameters to handle real-world trading environment.

Reference

- Binance Order Book API: https://developers.binance.com/docs/derivatives/option/market-data/Order-Book
- The Short-Term Predictability of Returns in Order Book Markets: a Deep Learning Perspective