Feature Engineering and Modeling for Short-Term Price Movement Prediction in ETH/USDT Using Tick-Level Data

Contents

1 Introduction

Cryptocurrency markets like ETH/USDT are characterized by high volatility and rapid price movements. Predicting price direction over short horizons (e.g., 1 minute) is of interest in high-frequency trading. This document provides a comprehensive pipeline that begins with tick-level data and results in a binary classifier for 1-minute price direction prediction.

2 Tick-Level Market Data

Tick data typically includes:

- **Bid Price** P_b: highest price a buyer is willing to pay.
- **Ask Price** P_a: lowest price a seller is willing to accept.
- **Bid Size** S_b: size of the top bid.
- **Ask Size** S_a: size of the top ask.
- **Trade Price** P_t: price at which a trade occurred.
- **Trade Size** V_t: size of the trade.
- **Side**: direction of aggressor (buy/sell).

2.1 Mid Price

We define the mid price as the average of the best bid and ask:

$$P_{\text{mid}} = \frac{P_b + P_a}{2}$$

The mid price serves as a stable reference for price movement prediction.

3 Feature Engineering

3.1 Order Book Imbalance (OBI)

Order book imbalance quantifies supply-demand pressure at the top of the book:

$$\text{OBI} = \frac{S_b - S_a}{S_b + S_a + \epsilon}$$

where ϵ is a small constant to avoid division by zero. Positive OBI indicates stronger buying pressure.

3.2 Trade Flow Imbalance (TFI)

TFI measures net market order pressure:

$$\text{TFI}(t) = \sum_{i \in [t-w,t]} V_i^{\text{buy}} - V_i^{\text{sell}}$$

for a rolling window w (e.g., 10 seconds), where

$$V_i^{\text{buy}} = \begin{cases} V_i & \text{if side = buy} \\ 0 & \text{otherwise} \end{cases} \quad \text{and} \quad V_i^{\text{sell}} = \begin{cases} V_i & \text{if side = sell} \\ 0 & \text{otherwise} \end{cases}$$

3.3 Volatility Clusters

We use the log return of mid prices:

$$r_t = \log\left(\frac{P_{\text{mid},t}}{P_{\text{mid},t-1}}\right)$$

$$\sigma_t = \sqrt{\frac{1}{N} \sum_{i=t-N+1}^{t} r_i^2}$$

Volatility clustering captures the heteroskedastic nature of price movements.

3.4 Microstructure Features

3.4.1 Spread

$$\text{Spread} = P_a - P_b$$

Tighter spreads indicate more liquid markets.

3.4.2 Signed Volume and Price Impact

Signed volume:

$$V^{\text{signed}} = \begin{cases} +V & \text{if side = buy} \\ -V & \text{if side = sell} \end{cases}$$

Price impact proxy:

$$\text{Impact} = \text{Spread} \times V^{\text{signed}}$$

3.4.3 Trade Intensity

Number of trades in a time window w:

$$\lambda_t = \sum_{i \in [t-w,t]} 1$$

4 Resampling and Aggregation

To model short-term movement, we aggregate features into 1-minute intervals. Aggregated statistics include:

- **OBI**: last value
- **TFI**: last value
- **Volatility**: last value
- **Spread**: mean
- **Price Impact**: mean
- **Intensity**: mean
- **Mid Price**: last value (used for target)

5 Label Construction

The target is the direction of price movement after 1 minute:

$$\text{Label}_t = \begin{cases} 1 & \text{if } P_{\text{mid},t+1} > P_{\text{mid},t} \\ 0 & \text{otherwise} \end{cases}$$

This is a binary classification problem.

6 Modeling

We use a Random Forest Classifier with the following settings:

- Number of estimators: 100
- Max depth: 5
- Training: 70% of data (time series split)

7 Evaluation Metrics

7.1 F1 Score

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

7.2 AUC-ROC

Area under the Receiver Operating Characteristic (ROC) curve:

$$\text{AUC} = \int_0^1 TPR(FPR^{-1}(x))dx$$

where TPR = True Positive Rate, FPR = False Positive Rate.

8 Python Implementation Summary

Key steps in Python:

1. Load and parse tick data.
2. Compute microstructure features.
3. Resample and label.
4. Train and evaluate a Random Forest model.

```
f1 = f1_score(y_test, y_pred)
auc = roc_auc_score(y_test, y_proba)
```

9 Feature Importance

Random Forest allows us to extract feature importances, showing which signals dominate:

$$\text{Feature Importance} \rightarrow \{\text{TFI}, \text{OBI}, \text{Volatility}, \dots\}$$

A bar plot can show the relative significance of each input.

10 Interpretation and Use in Trading

- High OBI and TFI can indicate an impending upward move.
- High volatility may indicate uncertain direction.
- Low spread and high intensity imply high liquidity periods.

11 Extensions

- Use LSTM for temporal dependencies.
- Add autocorrelation features.
- Incorporate economic or macro news triggers.
- Apply online learning models for live adaptation.

12 Conclusion

We showed how to:

- Engineer features from tick-level ETH/USDT data.
- Create a predictive model for 1-minute price direction.
- Evaluate with robust classification metrics.

This pipeline is a foundation for high-frequency crypto trading strategies.