Trading Bot

Raghav Goswami

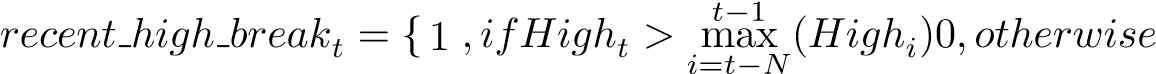
May 2025

# 1 Introduction

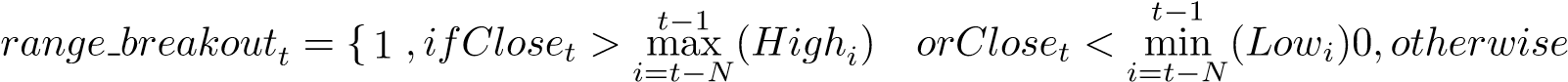
# Core Signal Logic

## Recent High Break

Let *N* be the lookback period (e.g., 20 bars). This signal fires when:



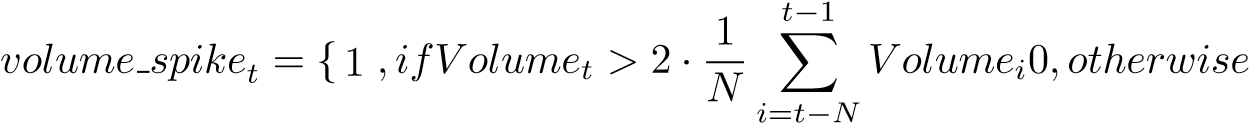
## Range Breakout



**Strong Candle**

*strong candlet* = {1 *,if*|*Closet* − *Opent*| *> ATRt*−10*,otherwise*

## Volume Spike



**RSI Bounce**

*rsi bouncet* = {1 *,ifRSIt*−1 *<* 45*andRSIt >* 450*,otherwise*

**MACD Cross Up**

*macd cross upt* = {1 *,ifMACDt*−1 *< Signalt*−1*andMACDt > Signalt*0*,otherwise*

## Match Score

A black background with a black square

AI-generated content may be incorrect.

**Final Signal**

*final signalt* = {1 *,ifmatch scoret* ≥ 40*,otherwise*

# Backtest Logic: MFE and MAE

Let *Pentry* be the entry price, and *ATRentry* the ATR on the entry bar. Let *Highj*, *Lowj* be the high and low during the open bars.

A black background with a black square

AI-generated content may be incorrect.

A black background with a black square

AI-generated content may be incorrect.

# Exit Logic (Intrabar TP/SL Simulation)

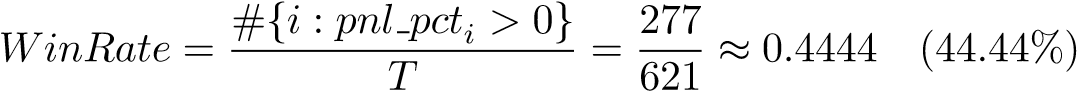
*IfHight* ≥ *TP price* ⇒ *ExitatTP*(*tp hit*)

*IfLowt* ≤ *SL price* ⇒ *ExitatSL*(*sl hit*)

# Key Metrics (Mathematical Breakdown)

Let *T* be the total number of trades. For our case, *T* = 621.

## 1. Win Rate



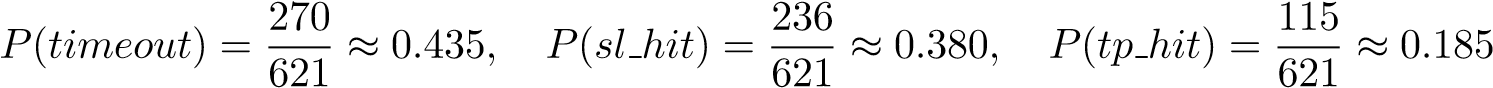
## 2. Average Return

A black background with a black square

AI-generated content may be incorrect.

*pnl pctwins* = 0*.*0995 (9*.*95%)*, pnl pctlosses* = −0*.*0735 (−7*.*35%)

## 3. Exit Reason Distribution



## 4. Holding Time

*duration candles* = 2*.*338*bars*

## 5. Signal Strength (match score)

*match score* = 4*.*399

|  |  |  |  |
| --- | --- | --- | --- |
| **match score** *k* | #Trades | Win Rate *k* | Avg pnl *k* |
| 4 | 375 | 45.87% | 0.529% |
| 5 | 244 | 41.80% | -0.073% |
| 6 | 2 | 100% | 15.34% |

Table 1: Performance by Match Score Buckets

# Entry Logic and Signal Definitions

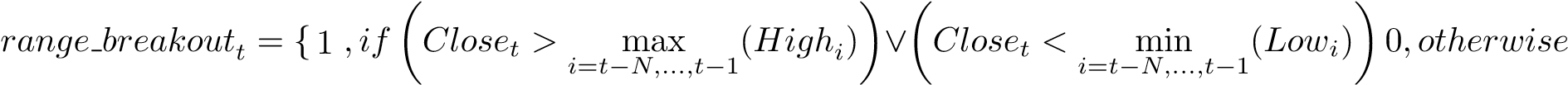
Let *N* be the lookback period (default: 20 bars).

## 1. Recent High Break

*recent high breakt* = {1 *,ifHight >* max (*Highi*)0*,otherwise*

*i*=*t*−*N,...,t*−1

## 2. Range Breakout



**3. Strong Candle**

*strong candlet* = {1 *,if* |*Closet* − *Opent*| *> ATRt*−10*,otherwise*

## 4. Volume Spike

*volume spiket* = {1 *,ifV olumet >* 2 · *V olumet*−*N*:*t*−10*,otherwise* Where:

A black background with a black square

AI-generated content may be incorrect.

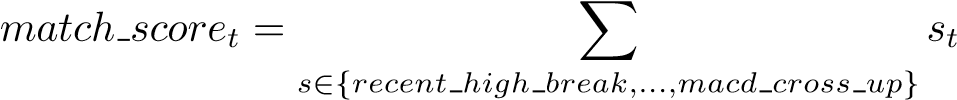
1. **RSI Bounce**

*rsi bouncet* = {1 *,if*(*RSIt*−1 *<* 45) ∧ (*RSIt >* 45)0*,otherwise*

1. **MACD Cross Up**

*macd cross upt* = {1 *,if*(*MACD histt*−1 *<* 0)∧(*MACD histt >* 0)0*,otherwise*

## 7. Match Score



## 8. Final Signal

*final signalt* = {1 *,ifmatch scoret* ≥ 40*,otherwise*

Backtest results

### Upgrade Plan — Final Checklist **Refine Signal Logic**

* Analyze signal combos from backtest results.
* Drop low-performing or one-off combos.
* **Update ALLOWED\_COMBOS** in run\_backtest\_v2() to:

ALLOWED\_COMBOS = {

"macd\_cross\_up+recent\_high\_break+strong\_candle+volume\_spike",

"macd\_cross\_up+range\_breakout+recent\_high\_break+strong\_candle+volume\_spike"

}

### **Add Debug Clarity**

* Add a logic\_debug\_note to each trade:

logic\_debug\_note = f"Score: {signal['match\_score']} | TP: {tp\_price:.2f} | SL: {sl\_price:.2f} | RSI: {df.iloc[i]['rsi']:.2f}"

Track which candle (1/2/3) hit TP/SL.

### **Entry Filters**

* Add optional ema\_trend\_confirmed = ema\_9 > ema\_20.
* Add minimum candle body size or ATR movement filter.

Backtest V2 Result Summary – ETH/USDT (1m candles)

**Bot Version:** V2  
**Exchange:** KuCoin  
**Data Points:** 10,000+ 1-minute candles

### **Key Performance Metrics**

* **Total Trades:** 779
* **Win Rate:** 43.5%
* **Avg Win %:** +0.18%
* **Avg Loss %:** -0.16%
* **Expectancy per Trade:** +0.01%
* **Profit Factor:** 1.10
* **Sharpe Ratio:** 0.29
* **Max Drawdown:** -6.94%
* **Avg Trade Duration:** 2.21 candles

Sharpe Ratio= (Average Return per Trade−Risk-Free Rate​)/Standard Deviation of Returns

* We assume **risk-free rate = 0**
* df['pnl\_pct'].mean() → average return per trade
* df['pnl\_pct'].std() → standard deviation of returns

sharpe\_ratio = df['pnl\_pct'].mean() / df['pnl\_pct'].std()

### **Max Drawdown**

**What it measures:**  
Largest peak-to-trough decline in cumulative returns — a key risk metric.

**How we calculated:**

Calculate cumulative return:

cum\_returns = (1 + df['pnl\_pct'] / 100).cumprod()

Track the rolling max of these cumulative returns:

roll\_max = cum\_returns.cummax()

Compute drawdown:

drawdown = (cum\_returns - roll\_max) / roll\_max

Max drawdown is:

max\_drawdown = drawdown.min()

### **3. Profit Factor**

**What it measures:**  
Ratio of gross profit to gross loss.

**Formula:**

Profit factor= sum of profitable trade returns/sum of losing trade returns

profit\_factor = df[df['pnl\_pct'] > 0]['pnl\_pct'].sum() / abs(df[df['pnl\_pct'] < 0]['pnl\_pct'].sum())

* If > 1: you make more on winning trades than you lose on losing ones.
* If < 1: your strategy loses money over time.

### **xpectancy**

**What it measures:**  
Average expected return per trade, accounting for both win rate and payout.

**Formula:**

Expectancy=(Pwin×Avg Win)+(Ploss×Avg Loss)\text{Expectancy} = (P\_{win} \times \text{Avg Win}) + (P\_{loss} \times \text{Avg Loss})Expectancy=(Pwin​×Avg Win)+(Ploss​×Avg Loss)

### expectancy = win\_rate \* avg\_win + (1 - win\_rate) \* avg\_loss**Win Rate**

**What it measures:**  
Proportion of profitable trades.

**Formula:**

Win rate = number of winning trades/total trades

win\_rate = len(df[df['pnl\_pct'] > 0]) / len(df)

### **Top Performing Signal Combos**

1. macd\_cross\_up + recent\_high\_break + strong\_candle + volume\_spike
   * 📈 **Avg PnL:** +0.021%
   * 📊 **Trades:** 55
   * ✅ **Win Rate:** 49%
2. macd\_cross\_up + range\_breakout + recent\_high\_break + strong\_candle + volume\_spike
   * 📈 **Avg PnL:** -0.005% (near break-even)
   * 📊 **Trades:** 291
   * ✅ **Win Rate:** 43%

### **Insights**

* Strategies with **volume confirmation and structure breakouts** performed better.
* Overcomplicated combos (4–5+ signals) often diluted the edge.
* Average profits remain small per trade — reflects realistic scalping conditions.
* MFE and MAE tracking shows decent risk-reward structure, but SL was often hit before TP.

### **Backtest Enhancements**

* Add atr\_on\_exit for volatility tracking.
* Add cooldown logic (e.g., skip next 3 candles after a loss).
* Group backtest results by signal\_combo\_name, match\_score for insights.

### **Actionable Improvements for V2.1**

* Restrict to top-performing signal combos only.
* Add RSI/EMA filters to reduce false entries.
* Add logic\_debug\_note to track decision-making per trade.
* Implement cooldowns to reduce overtrading in chop.

### **Trade Logic Flexibility**

* Make tp\_k and sl\_k dynamic based on signal strength:

if signal['match\_score'] >= 5:

tp\_k, sl\_k = 2.2, 1.2

### **Visual Analysis (optional if time permits)**

* Plot match\_score vs avg\_pnl\_pct
* Histogram of duration\_candles for profitable vs losing trades