Macroeconomic Event-Driven Strategy with Technical Analysis – QUIZ

Format: Jupyter Notebook (.ipynb)

Starting Capital: $10,000 USDT

Minimum Trades: 10

Objective

You are required to develop an intelligent trading strategy that anticipates and reacts to U.S. macroeconomic data releases. Your bot must analyze technical indicators before events and execute trades with strategic timing to maximize profitability.

Core Concept: Event-Driven Trading

Event-driven trading capitalizes on market volatility surrounding scheduled economic announcements. By analyzing technical indicators BEFORE these events, we can position ourselves to profit from anticipated market movements.

Entry & Exit Timing Strategy

ENTRY TIMING: Open positions 1, 2, or 3 days BEFORE macroeconomic announcements based on technical indicator signals. You must document which day you chose and why.

EXIT TIMING: Close positions 1, 2, or 3 days AFTER data release to capture market reaction. Your strategy should optimize this timing for maximum profit.

Macroeconomic Events (70 Total Releases)

Your strategy will track and trade around the following U.S. economic indicators:

1. **Unemployment Rate** - 10 releases (Monthly, High Impact)
2. **Nonfarm Payrolls (NFP)** - 10 releases (Monthly, Very High Impact)
3. **Continuing Jobless Claims** - 33 releases (Weekly, Medium Impact)
4. **Consumer Price Index (CPI)** - 10 releases (Monthly, Very High Impact)
5. **Fed Interest Rate Decision** - 7 releases (Quarterly, Extreme Impact)

Market & Data Specifications

- **Trading Pair:** BTC/USDT (Bitcoin vs US Dollar Tether)
- **Timeframe:** 1d (Daily candlesticks)
- **Data Source:** Binance MainNet API
- **Date Range:** January 1, 2025 - October 31, 2025
- **Starting Capital:** $10,000 USDT

- **Minimum Trades:** 10 executed trades

YOU MUST USE ONLY THE FOLLOWING 5 INDICATORS.

Using any other indicator will result in an automatic deduction. This restriction ensures that all students are evaluated on the same technical foundation and encourages deep understanding of these core indicators.

Required Indicators

1. SUPERTREND

- **Parameters:** period=10, multiplier=3

2. KAMA (Kaufman Adaptive Moving Average)

- **Parameters:** period=10, fast=2, slow=30

3. BOLLINGER BANDS

- **Parameters:** period=20, std_dev=2

4. RSI (Relative Strength Index)

- **Parameters:** period=14

5. MACD (Moving Average Convergence Divergence)

- **Parameters:** fast=12, slow=26, signal=9

Implementation Libraries

You may use any of the following Python libraries for indicator calculation:

```
# Option 1: TA (Technical Analysis Library)

import ta

from ta.trend import SuperTrendIndicator

from ta.volatility import BollingerBands

from ta.momentum import RSIIndicator

# Option 2: TA-Lib

import talib

# Option 3: Finta

from finta import TA

# Option 4: Pandas-TA

import pandas_ta as pta
```

BUY Signal Conditions

Entry Triggers (You can use one OR multiple conditions to trigger a BUY signal):

1. **Supertrend:** Trend direction is UP AND Price > Supertrend line
2. **KAMA:** Price crosses above KAMA (bullish crossover)
3. **Bollinger Bands:** Price < Lower Band (oversold condition)
4. **RSI:** RSI < 30 (oversold momentum)
5. **MACD:** MACD line > Signal line (bullish momentum crossover)

Signal Strategy Options

You have flexibility in how to use these signals:

Option A: Single Indicator Approach

- Use any ONE indicator that triggers to enter a trade
- Document which indicator you prioritize and why

Option B: Multiple Indicator Confirmation

- Require TWO or more indicators to confirm before entering
- Example: RSI < 30 AND MACD > Signal = Higher confidence entry
- Document your confirmation rules

Option C: Weighted Scoring System

- Assign scores to each indicator signal
- Enter trade when total score exceeds threshold
- Example: RSI<30 (3 points), MACD>Signal (2 points), threshold=4 points
- Document your scoring system

Important: Whatever approach you choose, document it clearly in your notebook!

IMPLEMENTATION REQUIREMENTS

Step 1: Data Collection

Objectives:

- Fetch BTC/USDT 1-day OHLCV data from Binance MainNet API
- Date range: 2025-01-01 to 2025-10-31
- Parse all 70 macroeconomic event dates into a DataFrame
- Handle weekends and market closures (use business days)
- Implement error handling for API failures

Deliverable: Two clean DataFrames - price data and event dates

Step 2: Indicator Calculation

Objectives:

- Calculate all 5 required technical indicators
- Handle NaN values appropriately (forward fill, interpolation)
- Validate indicator outputs for correctness
- Optimize parameters to generate minimum 10 trade signals
- Add indicator columns to the price DataFrame

Deliverable: Enhanced DataFrame with all indicator columns

Step 3: Trading Logic & Timing

Objectives:

- Implement flexible entry timing (1, 2, or 3 days before event)
- Implement flexible exit timing (1, 2, or 3 days after event)
- Generate trading signals based on indicator rules (single or multiple)
- Document your timing strategy and signal confirmation logic
- Ensure minimum 10 trades are executed
- Implement signal priority/scoring if using multiple indicators

Deliverable: Working signal generation function with documented strategy

Step 4: Backtesting

Objectives:

- Simulate trades for each macroeconomic event
- Apply entry and exit timing correctly
- Calculate profit/loss for each trade accurately
- Track cumulative balance throughout the period
- Implement position sizing (100% of available capital)
- Handle edge cases (no signal, weekend dates, etc.)

Deliverable: Complete backtesting results with all trades logged

Step 5: Reporting & Analysis

Objectives:

- Display results as a well-formatted DataFrame
- Include all required columns

- Calculate performance metrics (win rate, ROI, total P/L)
- Analyze timing performance (best entry/exit days)
- Generate summary statistics

Deliverable: Comprehensive results table and performance report

Common Pitfalls to Avoid

- **Look-ahead bias:** Don't use future data when making past decisions
- **Weekend handling:** Remember markets are closed on weekends
- **NaN values:** Indicators need warmup period, handle initial NaNs properly
- **Off-by-one errors:** Be careful with date arithmetic
- **Duplicate trades:** Ensure one trade per event maximum

TRADING RULES & TIMING STRATEGY

Entry Timing Strategies

You must decide HOW MANY days before each event to enter a position. Here are three approaches:

1. Fixed Strategy

- Always enter the same number of days before the event
- Example: Always enter 2 days before every event
- Pros: Simple, consistent
- Cons: Doesn't adapt to signal strength

2. Adaptive Strategy

- Adjust timing based on indicator strength or signal confidence
- Example:
    - Strong signal (RSI<25): 3 days before
    - Medium signal (RSI 25-30): 2 days before
    - Weak signal (RSI 30-35): 1 day before

3. Event-Type Strategy

- Different timing for different event types based on expected impact
- Example:
    - CPI/Fed Rate: 3 days before
    - NFP/Unemployment: 2 days before
    - Jobless Claims: 1 day before

Exit Timing Strategies

Similarly, decide HOW MANY days after the event to exit:

1. Fixed Exit

- Always exit the same number of days after the event
- Example: Always exit 1 day after to capture immediate reaction

2. Event-Based Exit

- Hold longer for high-impact events
- Example: CPI/Fed: 3 days, NFP: 2 days, Claims: 1 day

3. Profit Target Exit

- Exit when profit target reached or at maximum days
- Example: Exit if +3% profit reached, else wait max 3 days

Risk Management Rules

- **Position Size:** 100% of available capital (all-in per trade)
- **Stop Loss:** None (event-driven, we hold until exit date)
- **Take Profit:** Based on exit timing strategy
- **Maximum Trades:** One trade per event (no overlapping positions)
- **No Signal = No Trade:** If no indicator triggers, record as "HOLD"

Strategy Documentation Requirement

In your Jupyter Notebook, include a markdown cell explaining:

- Which timing strategy you chose (entry and exit)
- Which signal approach you used (single, multiple confirmation, or weighted)
- Why you chose this approach
- What you expected to achieve
- Any parameters you optimized

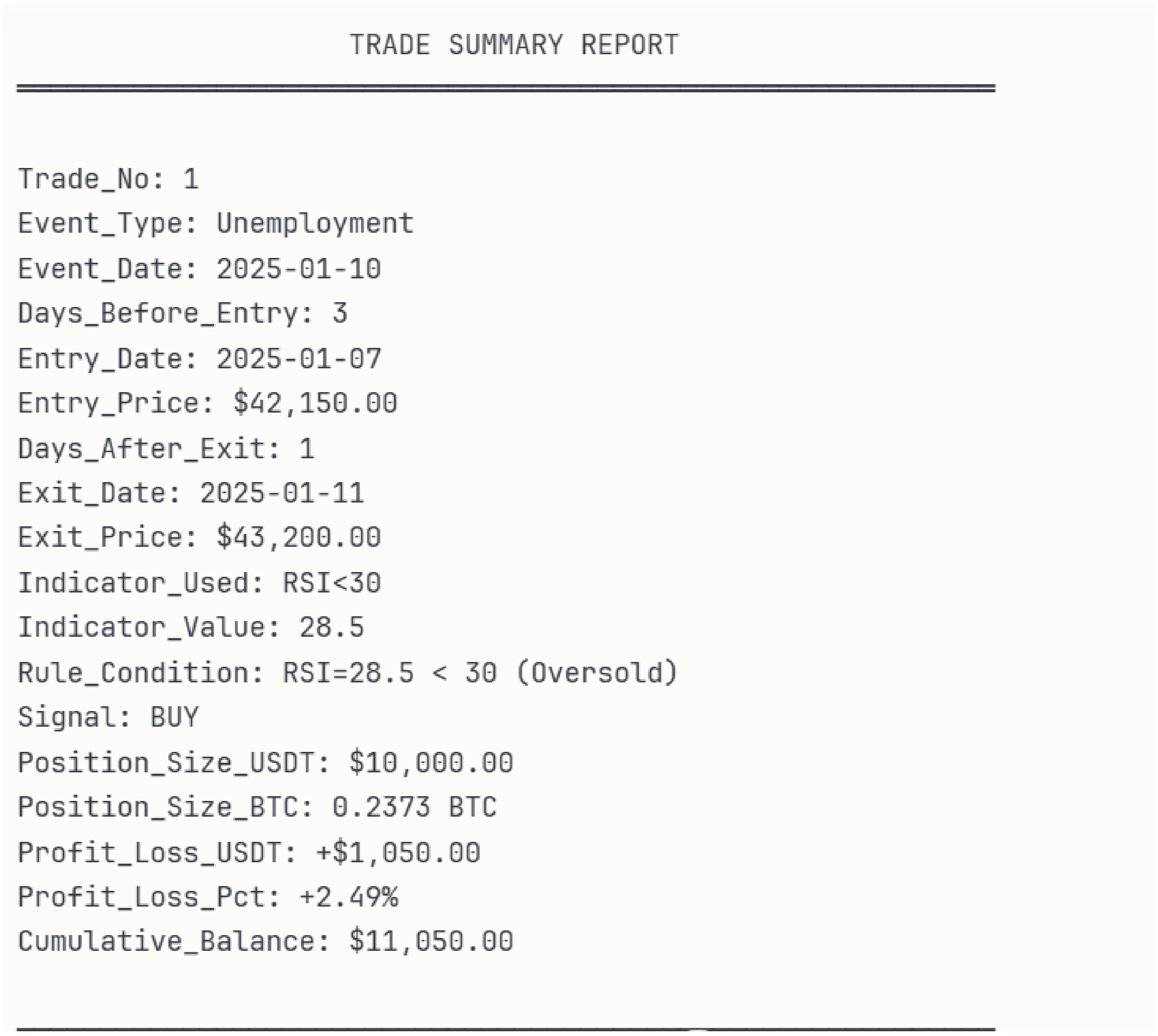
REQUIRED OUTPUT FORMAT

Complete Column Specifications

Your DataFrame MUST include these 18 columns:

1. **Trade_No** - Sequential trade number (1, 2, 3, ...)
2. **Event_Type** - Macroeconomic event category ("Unemployment", "CPI", "Fed_Rate")
3. **Event_Date** - Official data release date (2025-01-10)

4. **Days_Before_Entry** - How many days before event you entered (1, 2, or 3)
5. **Entry_Date** - Actual position open date (2025-01-07)
6. **Entry_Price** - BTC price at entry in USDT (42150.50)
7. **Days_After_Exit** - How many days after event you exited (1, 2, or 3)
8. **Exit_Date** - Actual position close date (2025-01-11)
9. **Exit_Price** - BTC price at exit in USDT (43200.00)
10. **Indicator_Used** - Which indicator(s) triggered the signal ("RSI<30", "RSI+MACD")
11. **Indicator_Value** - The indicator's value at entry (28.5 for RSI, or "RSI:28.5, MACD:125.3" for multiple)
12. **Rule_Condition** - Detailed explanation of the rule ("RSI=28.5 < 30 (Oversold)")
13. **Signal** - Trade decision ("BUY" or "HOLD")
14. **Position_Size_USDT** - Investment amount in dollars (10000.00)
15. **Position_Size_BTC** - BTC quantity purchased (0.2373)
16. **Profit_Loss_USDT** - Profit or loss in dollars (+1050.00 or -500.00)
17. **Profit_Loss_Pct** - Profit or loss percentage (+2.49% or -1.19%)
18. **Cumulative_Balance** - Total account balance after trade (11050.00)



```
                    TRADE SUMMARY REPORT
════════════════════════════════════════════════════════════


Trade_No: 1
Event_Type: Unemployment
Event_Date: 2025-01-10
Days_Before_Entry: 3
Entry_Date: 2025-01-07
Entry_Price: $42,150.00
Days_After_Exit: 1
Exit_Date: 2025-01-11
Exit_Price: $43,200.00
Indicator_Used: RSI<30
Indicator_Value: 28.5
Rule_Condition: RSI=28.5 < 30 (Oversold)
Signal: BUY
Position_Size_USDT: $10,000.00
Position_Size_BTC: 0.2373 BTC
Profit_Loss_USDT: +$1,050.00
Profit_Loss_Pct: +2.49%
Cumulative_Balance: $11,050.00

────────────────────────────────────────────────────────────
```

```
Trade_No: 2
Event_Type: CPI
Event_Date: 2025-01-15
Days_Before_Entry: 2
Entry_Date: 2025-01-13
Entry_Price: $43,800.00
Days_After_Exit: 2
Exit_Date: 2025-01-17
Exit_Price: $44,500.00
Indicator_Used: RSI+MACD (Multiple Confirmation)
Indicator_Value: RSI:29.3, MACD:125.3
Rule_Condition: RSI=29.3<30 AND MACD=125.3>Signal=118.7
Signal: BUY
Position_Size_USDT: $11,050.00
Position_Size_BTC: 0.2523 BTC
Profit_Loss_USDT: +$700.00
Profit_Loss_Pct: +1.60%
Cumulative_Balance: $11,750.00
```

```
═══════════════════════════════════════════════════════════
                      FINAL RESULTS
═══════════════════════════════════════════════════════════
Total Trades:                15
Successful Entries:          12 (with BUY signal)
No Signal (HOLD):            3
Winning Trades:              9
Losing Trades:               3
Win Rate:                    75.0%
Average Days Before Entry:   2.1
Average Days After Exit:     1.8
Total Profit/Loss:           +$3,240.00
Final Balance:               $13,240.00
ROI:                         32.4%
═══════════════════════════════════════════════════════════
```

Additional Performance Metrics

Your output should also calculate and display:

Basic Metrics:

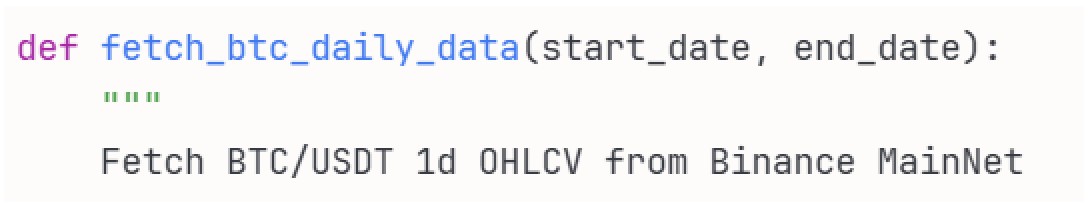
- Total number of trades
- Number of winning trades
- Number of losing trades
- Win rate (%)
- Total P/L (USDT)
- ROI (%)

Advanced Metrics:

- Average P/L per trade
- Best trade (highest profit)
- Worst trade (largest loss)
- Average entry timing (days before)
- Average exit timing (days after)
- Most profitable indicator (or indicator combination)

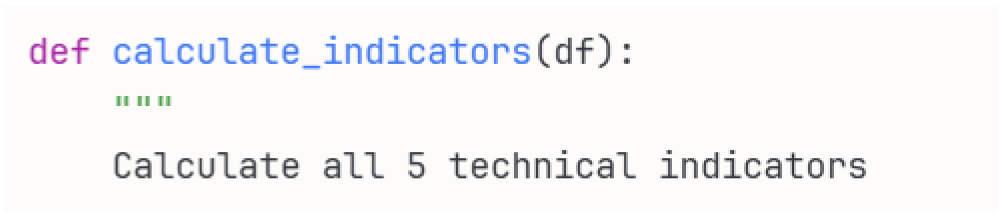
CODE STRUCTURE

1. DATA COLLECTION



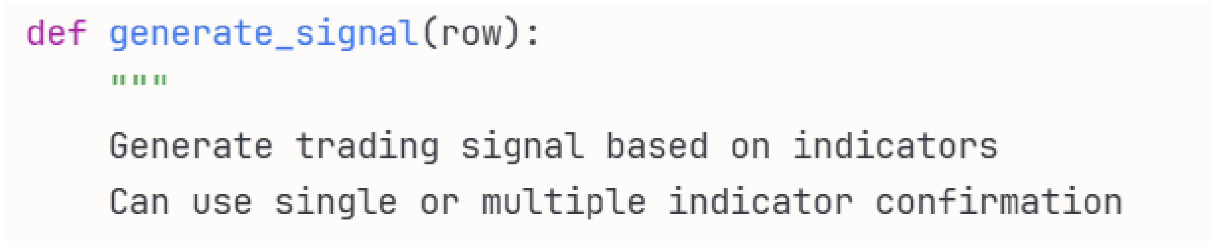
```python
def fetch_btc_daily_data(start_date, end_date):
    """
    Fetch BTC/USDT 1d OHLCV from Binance MainNet
```

2. INDICATOR CALCULATION

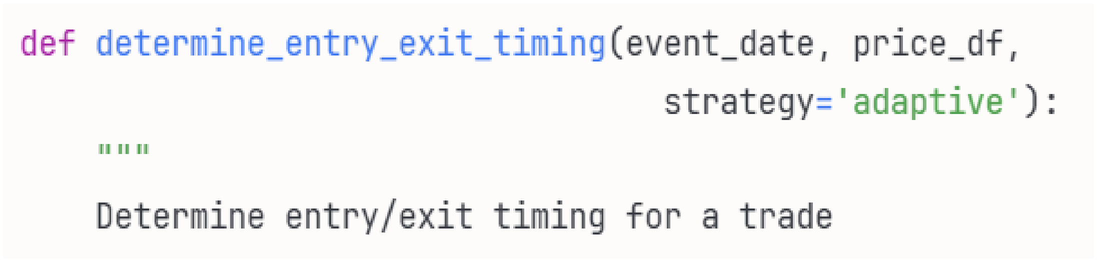


```python
def calculate_indicators(df):
    """
    Calculate all 5 technical indicators
```

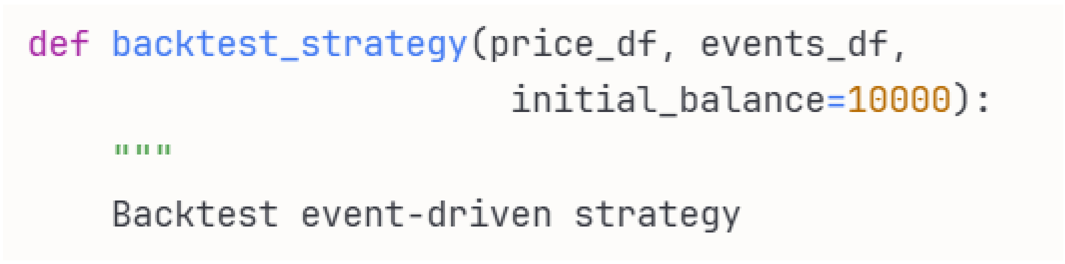
3. TRADING LOGIC



```python
def generate_signal(row):
    """
    Generate trading signal based on indicators
    Can use single or multiple indicator confirmation
```



```python
def determine_entry_exit_timing(event_date, price_df,
                                strategy='adaptive'):
    """
    Determine entry/exit timing for a trade
```



```python
def backtest_strategy(price_df, events_df,
                      initial_balance=10000):
    """
    Backtest event-driven strategy
```

Primary File: Jupyter Notebook (.ipynb)

- File name: YourName_BTC_Trading_Quiz.ipynb

FREQUENTLY ASKED QUESTIONS

Q1: What if there's no trading signal on my calculated entry date? A: Record it as Signal: "HOLD" in your DataFrame. No trade is executed, balance remains unchanged, but you still document the event. This counts toward your analysis but not toward your 10 minimum trades.

Q2: Can I use partial positions instead of 100% capital? A: Yes, for advanced implementation. You can adjust position size based on signal strength (e.g., 50% for weak signals, 100% for strong). Just document your approach clearly and ensure it's consistent.

Q3: How do I handle NaN values in indicators? A: Indicators need a warmup period. For the first 14-20 rows, you'll have NaN values. Use df.fillna(method='ffill') or skip those rows when generating signals. Never use future data to fill past NaNs!

Q4: What if I can't get exactly 10 trades with default parameters? A: Adjust your indicator thresholds:

- RSI: Try < 35 instead of < 30
- Bollinger Bands: Use 2.5 std dev instead of 2
- MACD: Reduce sensitivity Document why you made these changes!

Q5: Should I use single or multiple indicator confirmation? A: Either approach is valid!

- Single indicator: Simpler, more trades, may have more false signals
- Multiple confirmation: Fewer but higher quality trades, may miss opportunities Choose based on your strategy and document your reasoning!

Q6: How do I document multiple indicator signals? A: In your output columns:

- Indicator_Used: "RSI+MACD" or "RSI+BB+MACD"
- Indicator_Value: "RSI:28.5, MACD:125.3"
- Rule_Condition: "RSI=28.5<30 AND MACD=125.3>Signal=118.7"