A Beginner-Friendly Cryptocurrency Trading Bot in Python.

Crypto Pilot is a simple, educational trading bot that demonstrates basic crypto trading strategies and automated alerts. Built as a learning project, it's designed to explore trading logic, risk management, and backtesting in Python.

Version 1

Amin Hydar Ali

Let’s learn and build together !

🌟 Got any ideas, suggestions, or want to collaborate?

Reach out anytime: aminhydarali@gmail.com

Check out the repo:
https://github.com/alaminhydar/Crypto-Pilot/

CryptoPilot is a paper-trading framework for crypto futures that downloads OHLCV data from an exchange (Bitget via `ccxt`), computes indicators (EMA, RSI, MACD, ATR), converts indicator signals into trade entries/exits with a trend filter, simulates trade execution and margin via a `PaperBroker`, uses ATR-based stop-loss / take-profit and ATR-based position sizing, logs every trade to CSV and sends short Telegram alerts for events. It is intentionally **paper-only** (no real orders). The pipeline is:

fetch data → compute indicators & signals → simulate trades → log & alert → final summary & metrics.

What This Project Actually Is (Explained Step-by-Step)

This project is a **paper-trading system** for crypto futures.
That means:

> It **pretends** to trade the crypto market using real historical or live price data, but **without using real money** yet. It's a safe environment to test strategies.

Let's break the description into parts:

1. "Paper-trading trading system for crypto futures"

- **Paper trading** = simulated trading.
  You trade as if using real money, but no real funds are used.
  It's like "practice mode".
- **Crypto futures** = a type of trading where you can profit from **both rising and falling prices**, and often with leverage.
  Example:
  - If you think Bitcoin will rise → you open a **long**

    - If you think Bitcoin will fall ⟶ you open a **short**

This project focuses on testing strategies for futures trading, but **without executing real trades yet**.

2. “It downloads historical price data”

The bot first needs past market information to analyze and generate signals.

- It uses **OHLCV** data:
    - **O**pen
    - **H**igh
    - **L**ow
    - **C**lose
    - **V**olume

This comes from the exchange (Bitget) using the **ccxt** library.
Historical data is required to:

- calculate indicators
- test the strategy on past charts
- simulate trading decisions

3. “Computes technical indicators (EMA, RSI, MACD, ATR)”

It uses math-based tools called **indicators** to understand the trend, momentum, and volatility.

Indicator	Full Name	Purpose
EMA	Exponential Moving Average	Identifies trend direction and momentum shifts
RSI	Relative Strength Index	Measures overbought/oversold, momentum strength
MACD	Moving Average Convergence Divergence	Confirms trend changes and momentum direction

Indicator	Full Name	Purpose
ATR	Average True Range	Measures volatility to size trades and set SL/TP

The bot uses these to decide when to buy or sell.

4.“Generates buy/sell signals with a trend filter”

The strategy doesn’t just randomly buy or sell, it follows rules based on the indicators.

A **signal** is a moment when the strategy says:

- “conditions are right, **buy now**”
- or“momentum is reversing, **sell now**”

A **trend filter** (like the 200 EMA) prevents trading against the overall trend to avoid losing trades.

Example logic:

- If the market is above the 200 EMA → only look for **buys**
- If below the 200 EMA → only look for **sells**

This greatly improves accuracy.

5.“Simulates trade execution (paper broker)”

The system includes a virtual broker called `PaperBroker`.
Its job is to **act like an exchange**:

It tracks:

- balance
- open positions
- entry price
- stop-loss
- take-profit

When a signal appears:

- It "opens a trade"
- Deducts the cost from the virtual balance
- Monitors the trade until it closes

No real money is touched, this is **simulation only**.

6. "Using ATR-based stop-loss / take-profit and ATR-based position sizing"

This is about **risk management**.

• **Stop-loss** = entry_price − (ATR × SL_multiplier)
• **Take-profit** = entry_price + (ATR × TP_multiplier)
• **Position size** = number of units so that 1×ATR move risks risk_pct of account

ATR is used because it adjusts SL/TP to market volatility:

- If market is very volatile ⟶ wider SL/TP
- If market is calm ⟶ tighter SL/TP

ATR-based position sizing means:

> Trade size is calculated based on volatility so you risk a consistent % per trade (e.g., 1% of your account).

This ensures you don't risk too much on volatile markets or too little on calm markets.

7. "Logs events"

Every trade (buy, sell, SL, TP) is recorded.
This is useful for:

- debugging
- performance analysis

- backtesting audits
- improving the strategy later

📢 8. "Sends Telegram alerts"

The bot can send notifications to Telegram whenever:

- a trade opens
- stop-loss hits
- take-profit hits
- paper-trading finishes

This is how you can monitor the bot without staring at the computer.

9. "It does not place real orders (that's for a later, production step)."

Important for beginners:

Right now, the bot **only simulates trades**.
It does *not* send real buy/sell orders to the exchange.

This is intentional, to:

- avoid losing money while testing
- improve strategy first
- collect data to confirm profitability

Later, we will upgrade it to a **live trading bot** by adding:

- Order execution via exchange API
- Websocket price feeds
- Error handling
- Risk limits
- Anti-liquidation logic

What is a Moving Average (MA)?

- Imagine the stock market or crypto market like a rollercoaster of prices: it goes up, down, up, down… all the time.
- Looking at the raw prices can be confusing, because they jump a lot every second or minute.

Moving Average is like smoothing out the bumps in that rollercoaster so you can see the overall trend.

- Think of it like walking through a bumpy road: instead of focusing on every small stone, you look at the road as a smooth path.
- In trading, the moving average gives you a "smoothed line" that shows whether the price is generally going up, down, or sideways.

Simple Example of Moving Average

- Suppose Bitcoin prices over 5 days were: 100, 105, 103, 108, 110
- Simple Moving Average (SMA): Take the average of the last N days.

Example: 3-day SMA:

- Day 3 ⟶ (100+105+103)/3 = 102.67
- Day 4 ⟶ (105+103+108)/3 = 105.33
- Day 5 ⟶ (103+108+110)/3 = 107

This gives a smooth line instead of jumping from 100 ⟶ 105 ⟶ 103 ⟶ 108 ⟶ 110.

What is Exponential Moving Average (EMA)?

- EMA is a special kind of moving average that gives more weight to recent prices.

- Why? Because the most recent price usually matters more than what happened a month ago.

Analogy:

- Imagine you're tracking your weight every day.

- You want to see recent trends, not just the average of the past 6 months.

- EMA says: "today's weight is more important than your weight 5 months ago."

- Think of it like memory that forgets old prices gradually but remebr recent ones more clearly.

- A 1-day EMA reacts faster to recent price changes than a simple moving average (SMA).

EMA Math (simplified)

- Each new price affects the EMA more than older prices.

- Formula (simplified idea):
  $\textbf{EMA today}=(\textbf{Price today}*\mathbf{k})+(\textbf{EMA yesterday}*(\mathbf{1-k}))$

where:

Price_today: the current price of the asset.
EMA_yesterday: the EMA value from the previous day (or previous period).

Where:

- k = weight of today's price
- The bigger the k → EMA reacts faster to price changes
- The smaller the k → EMA is smoother, slower to react

Example:

- Let's say we want a 3-day EMA.
- Price for the last 3 days: 10, 11, 12
- First EMA (EMA_yesterday) = first_price = 10
- smoothing factor:
  K = 2/(3 + 1) = 0.5

Day 2 EMA:

EMA = (11 x 0.5) + (10 x 0.5) = 5.5 + 5 = 10.5

Day 3 EMA:
EMA = (12 x 0.5) + (10.5 x 0.5) = 6 + 5.25 = 11.25

And so on, EMA reacts to faster recent prices than a simple average.

3 EMA in the Market

- Traders use EMA to identify trends.

Examples:

1. **Price above EMA → Uptrend**
    - If Bitcoin is trading above the EMA, it means the market is generally moving up.
    - Analogy: You're climbing a hill, the slope is positive.
2. **Price below EMA → Downtrend**
    - If price is below the EMA, it signals a potential downtrend.
    - Analogy: You're going downhill.
3. **EMA crossovers**
    - Traders often use two EMAs: a short-term (fast) EMA and a long-term (slow) EMA.
    - Example:
        - 8-period EMA (fast)
        - 21-period EMA (slow)
    - Buy Signal: Fast EMA crosses above slow EMA → momentum turning up
    - Sell Signal: Fast EMA crosses below slow EMA → momentum turning down

4 Analogy for EMA

- Imagine you are tracking the speed of a car:
    - **SMA**: looks at average speed over the last 1 hour equally
    - **EMA**: cares more about the last 5 minutes than 55 minutes ago
- EMA reacts faster to recent acceleration or braking.

5 Why EMA is Popular

- Reacts faster to price changes than SMA
- Helps traders catch trends earlier
- Works well with other indicators (RSI, MACD)
- Very useful in crypto trading, where prices move fast

6 Visual Example (Imagine a Chart)

- Bitcoin chart shows candles moving up and down
- Draw a 21 EMA (fast) ⟶ smooth line hugging the price, reacts quickly
- Draw a 50 EMA (slow) ⟶ smoother line, slower
- When the fast EMA crosses the slow EMA ⟶ often a signal to buy or sell

The market basics

- **OHLCV** stands for **Open, High, Low, Close, Volume** for each candle (time period).
    - Example for one 1-hour candle:
        - **open**: price at the start of the hour (e.g., 37,000)
        - **high**: highest price during the hour (e.g., 37,200)
        - **low**: lowest price during the hour (e.g., 36,800)
        - **close**: price at the end of the hour (e.g., 37,050)
        - **volume**: amount traded during the hour (e.g., 120 BTC)
- **Why candles?** They summarize price action in discrete time slices so indicators and strategies can be computed.
- **Volatility** = how much price moves. High volatility means big up/down swings; low volatility means small moves.
- **ATR (Average True Range)** measures typical candle movement (volatility). Higher ATR ⟶ wider expected swings.

What each dependency / import does (so you know why it's there)

- **ccxt**: A standard library that connects to many crypto exchanges (Bitget, Binance, etc.). We use it to fetch historical OHLCV.
- **pandas (imported as pd)**: The core data tool. Works like an Excel spreadsheet in code. Stores candle rows, computes columns, slices data, saves logs.
- **ta**: A technical-analysis helper library. Computes RSI, MACD, ATR, and other indicators with correct formulas and smoothing.
- **matplotlib.pyplot**: (used for plotting) creates charts (equity curve, price + indicators).

- **python-dotenv / os.getenv:** load environment variables (API keys) so keys are not hard-coded.
- **telegram / telebot**: send messages to Telegram for alerts.
- **typing.Optional:** convenience type hints (not necessary to run).

Files and responsibilities (module map)

- **data/data.collector.py**
  - Connects to Bitget via ccxt.
  - **init_bitget(...)** returns an exchange object.
  - **fetch_ohlcv(...)** calls **exchange.fetch_ohlcv(...)** and returns a pandas DataFrame with columns **['timestamp','open','high','low','close','volume'].**
- **strategies/strategies.py**
  - **ema_rsi_macd(df, ...)** computes:
    - 200 EMA (trend filter),
    - short EMA (8), long EMA (21),
    - RSI (14),
    - MACD and MACD signal,
    - sets signal column: 1 = buy signal, −1 = sell signal,
    - sets position = change in signal (so it marks entries/exits).

1. 200 EMA – Trend Filter

What it does

Calculates the **200-period Exponential Moving Average**.

Why it matters

The 200 EMA is a **macro trend filter** used by professionals. It tells you the dominant long-term trend of the market.

Price vs 200 EMA	Interpretation
• Price **above** 200 EMA	• Uptrend - only look for buys
• Price **below** 200 EMA	• Downtrend - only look for sells

This prevents taking trades against the dominant trend and **eliminates many losing trades**.

2. Short EMA (8) & Long EMA (21)

What it does

Computes 8 EMA and 21 EMA.

Why it matters

These fast-moving EMAs help detect **short-term momentum shifts**.

- When **8 EMA crosses above 21 EMA** → bullish momentum
- When **8 EMA crosses below 21 EMA** → bearish momentum

This is essentially a **momentum-based entry system**.

3. RSI (14)

What is RSI?

- RSI is a momentum indicator that tells us whether an asset (like Bitcoin) is overbought or oversold.
- Think of it as a speedometer for price momentum: how fast the price is rising or falling.
- Momentum = **how fast the price is moving and in which direction**

- RSI measures whether an asset is **“moving up too fast” or “moving down too fast”** over a set period (default = 14 candles).

Think of RSI as a **speedometer for price movement**:

- 0 ⟶ extremely slow / oversold ⟶ sellers exhausted
- 50 ⟶ neutral ⟶ balanced market
- 100 ⟶ extremely fast / overbought ⟶ buyers exhausted

Why use RSI?

- To find potential reversals:
    - Overbought ⟶ price might drop soon
    - Oversold ⟶ price might rise soon
- **Helps traders avoid buying at the top or selling at the bottom.**

How is RSI calculated?

1. Look at the **average gain** and **average loss** over a period (usually 14 periods).
    - Gain = amount price went up
    - Loss = amount price went down
2. Compute **RS = Average Gain / Average Loss**
3. Compute **RSI = 100 - (100 / (1 + RS))**

RSI ranges from **0 to 100**:

- Above 70 ⟶ Overbought (price might fall soon; caution before buying)
- Below 30 ⟶ Oversold (price might rise soon; caution before selling)

Note: In our bot, we use 65 as the overbought threshold for more conservative entries.

How RSI works

RSI compares **average gains** to **average losses** over the last N periods:

$$RSI=100-1+RS/100$$

Where:

RS=Average Loss over N periods / Average Gain over N periods

- **Average Gain** = how much price increased on up days
- **Average Loss** = how much price decreased on down days

Step-by-step

1. Look at the last N periods (e.g., 14 candles)
2. Separate upward moves and downward moves
3. Calculate the average gain and average loss
4. Compute RS = Avg Gain ÷ Avg Loss
5. Map it to 0–100 scale ⟶ RSI

How it detects overbought or oversold

- **Overbought** ⟶ **RSI > 70** (default threshold)
    - Price has risen a lot ⟶ buyers may be exhausted
    - Market could **reverse or consolidate**
- **Oversold** ⟶ **RSI < 30** (default threshold)
    - Price has fallen a lot ⟶ sellers may be exhausted
    - Market could **bounce back or reverse**

Example

Suppose 14 candles have the following moves:

- 9 up-days averaging +2
- 5 down-days averaging −1

RS= 2 / 1 =2
RSI = 100 – 100 /1+2 = 100 − 100 / 3 = 66.7

- **RSI ≈ 66** ⟶ not overbought yet (needs > 70)
- **RSI < 70** ⟶ safe to buy (in this bot, we use <65 as conservative threshold)
- RSI **looks back in time** (last N candles)
- Compares **upward vs downward strength**
- Maps this comparison to 0–100
- Thresholds tell you if the market is**"over-excited""** in either direction
- Market = a race
- RSI = referee measuring how fast runners are going uphill or downhill
- If runners are sprinting too fast ⟶ referee warns (overbought)
- If runners are crawling too slowly ⟶ referee says market is"ready to bounce" (oversold)

Why default 14 periods?

- **14 is an industry standard** (introduced by J. Welles Wilder, RSI creator)
- It balances **responsiveness vs. smoothness**:

  - Too short (like 5 periods) ⟶ RSI moves too wildly ⟶ false signals
  - Too long (like 50 periods) ⟶ RSI reacts too slowly ⟶ signals come late
- 14 periods = good middle ground for most markets, works for daily, hourly, or even intraday charts

Analogy:

- **RSI = a thermometer for market "temperature":**
  - 14-period RSI = "average temperature over last 14 candles" ⟶ smooth enough to avoid noise but fast enough to notice trends
  - 5-period RSI = "temperature every 5 minutes" ⟶ jumps up and down too quickly
  - 50-period RSI = "temperature every 50 minutes" ⟶ slow to notice changes

How timeframe affects RSI

- RSI measures **momentum relative to its own timeframe**:
  - Daily chart ⟶ RSI looks at **14 days of price changes**
  - 1-hour chart ⟶ RSI looks at **14 hours of price changes**
  - 15-min chart ⟶ RSI looks at **14 × 15-min candles (~3.5 hours)**
- The RSI's "overbought" and "oversold" readings are **relative to the timeframe**, not absolute price levels

Example: Price over 7 candles (shortened for simplicity)

Assume closing prices for 7 consecutive candles:

Candle	Close Price
1	100
2	102
3	101
4	103
5	105
6	104
7	107

Step 1: Calculate gains and losses

- Gain = price went **up** from previous candle
- Loss = price went **down** from previous candle

Candle	Close	Change	Gain	Loss
1	100	—	—	—
2	102	+2	2	0
3	101	-1	0	1
4	103	+2	2	0
5	105	+2	2	0
6	104	-1	0	1
7	107	+3	3	0

Step 2: Calculate average gain and loss

- For simplicity, let's just take the **simple average** over these 6 changes (normally RSI uses **smoothed average**)

  Avg Gain = 2+0+2+2+0+3 / 6 = 9/ 6 = 1.5
  Avg Loss = 0+1+0+0+1+0 / 6 = 2 / 6 ≈ 0.333

Step 3: Calculate RS (Relative Strength)

RS = Avg Gain / Avg Loss ≈ 4.5

Step 4: Calculate RSI

RSI = RSI = 100 − 100 / (1 + 4.5) = 100 − 100 / 5.5 ≈ 81.8

RSI ≈ 82 ⟶ overbought territory (above 70)

Step 5: Understanding

- The bot sees this as:

  > "Prices have been rising faster than they have been falling over the last 7 periods ⟶ buyers may be exhausted ⟶ caution before buying more."

- RSI tells the bot **not to buy if RSI is above 65 (our bot's threshold)**, even if other indicators are bullish.

1. Why not buy when RSI is ~65–70?

- **RSI measures momentum** - it tells you how strong recent price gains are compared to losses.
- When RSI is **high (above 65–70)**:
  1. Prices have been rising **consistently** over recent periods.
  2. Most buyers have already entered ⟶ there are **fewer new buyers left.**
  3. Market is **overextended** ⟶ high chance of a **pullback or consolidation**.
- Buying when RSI is high = "buying at the top" ⟶ risky because the **momentum might soon reverse**.

Analogy:

- Imagine a crowded rollercoaster ride:
  - Everyone is rushing to get on ⟶ the ride is almost full.

    - If you jump in now, you’re more likely to **wait in line or get stuck**.
    - RSI ~65–70 = ride is almost full ⟶ best to wait for a safer entry.

2. How RSI arrives at that number

- RSI = 100 − 100 / (1 + RS)
- RS = **Average Gain ÷ Average Loss** over N periods

Step-by-step example:

1. Suppose last 14 candles: prices mostly **went up** ⟶ **average gain > average loss**
2. Calculate RS:

   RS = Avg Gain / Avg Loss

- Example: **Avg Gain = 1.5, Avg Loss = 0.5 ⟶ RS = 3**

3. Calculate RSI:

   RSI = 100 – 100 / 1 + RS = 100 − 100 / 1 = 3 = 100 − 25 = 75

- **RSI = 75** ⟶ market **overbought**

So the number itself **comes directly from the ratio of recent gains vs losses**.

3. Why our bot uses 65 instead of 70

- Traders often **lower the threshold slightly** (from 70 ⟶ 65) to be **more conservative**
- Ensures the bot **avoids buying too late**, before the market reaches extreme overbought levels
- Acts as **early warning** instead of waiting for absolute overbought

4. Analogy

- **RSI 50** → market balanced → buyers = sellers
- **RSI 65** → momentum strongly favors buyers → but market is **getting crowded**
- **RSI 70+** → market extremely crowded → high probability of pullback

So the bot treats RSI < 65 as a green light to buy, otherwise it waits.

Why it matters

RSI measures momentum speed. It helps filter fake signals given by EMAs alone.

RSI Value	Interpretation
> 50	Bullish momentum
< 50	Bearish momentum
> 70	Overbought (avoid buying, or prepare to sell)
< 30	Oversold (avoid selling, or prepare to buy)

So often you don't buy unless RSI > 50 (confirm momentum).

Best time to buy using RSI

- RSI is not the only factor, it works best combined with trend/momentum indicators (like EMAs, MACD in your bot)
- Typical "good buy" scenario in our bot:
  1. Price above 200 EMA → market in uptrend
  2. Short-term momentum confirmed → 8 EMA > 21 EMA
  3. RSI < 65 → market still has room to rise → not overbought

4. MACD confirms momentum → MACD > Signal line

When all 4 conditions are true → **high-probability buy**

MACD (12, 26, 9) and MACD Signal

MACD means Moving Average Convergence Divergence.

Let's break the words down so it's not just a definition but a clear understanding:

Moving Average

A moving average smooths price to show the trend.
MACD uses **Exponential Moving Averages (EMAs)**, which react faster to price.

🔄 Convergence

This means **the two EMAs are coming closer together**.
When they converge, it shows the momentum is weakening or trend is changing.

↔️ Divergence

This means **the two EMAs are moving further apart**.
When they diverge, it shows momentum is strengthening and a trend is growing.

Putting it Together

MACD = A tool that measures how the 12 EMA and 26 EMA are converging or diverging to show trend momentum.

- If 12 EMA goes far above 26 EMA → Strong bullish momentum (divergence ↑)

- If 12 EMA moves closer to 26 EMA ⟶ Momentum weakening (convergence)

The MACD line visualizes this relationship.

MACD tells you how strong a trend is and when that trend is changing, by measuring the distance between two EMAs.

What it does

- MACD is another **momentum and trend indicator**.
- It tells us **how fast the trend is moving and when it might change direction**.
- Unlike RSI, MACD is based on **moving averages**.

Why it matters

MACD is a trend + momentum + reversal indicator.
It confirms whether a trend shift is real or a fake breakout.

MACD Crossover	Meaning
MACD above signal	Bullish slope
MACD below signal	Bearish slope

Used as an extra confirmation layer to avoid false EMA crosses.

Combining EMA values does NOT automatically create MACD

Just because you use **two EMAs together (like 8 & 21, or 12 & 26, or 50 & 200)** does **NOT** mean you now have MACD.

Why?

MACD is calculated as:
• MACD Line = 12 EMA – 26 EMA
• Signal Line = 9 EMA of the MACD Line
Just using two EMAs (like 8 & 21) is only a crossover strategy, not MACD.

They only become **MACD** after you do the MACD formula.

Difference Between Using 8/21 EMA and MACD

8/21 EMA Strategy

You're comparing two EMAs:

- When 8 EMA crosses above 21 EMA → Buy signal
- When 8 EMA crosses below 21 EMA → Sell signal

This is **just a crossover strategy**.

MACD Strategy

MACD calculates momentum by subtracting EMAs:

MACD Line = 12 EMA − 26 EMA
Then MACD creates a **Signal Line = 9 EMA of MACD Line**

This measures:

✔️ Trend direction
✔️ Speed of trend
✔️ Momentum shifts
✔️ Possible reversals

📍 MACD = 3 Numbers: 12, 26, 9, But Why These?

MACD was created by **Gerald Appel in the 1970s**. He tested many combinations of moving averages on stocks and discovered:

- 12 and 26 worked best for detecting momentum change
- 9 worked best for smoothing the signal to reduce noise

Meaning of the Numbers:

Number	Meaning	Role
12	Short-term price movement	Fast EMA

26	Medium-term movement	Slow EMA
9	Smooth signal	EMA applied to MACD line

Why 9 for the Signal Line?

Because **MACD Line can be too noisy** (too many false signals).
So Appel applied a **9-period EMA** to the MACD Line to create a smoother line.

This helps traders avoid reacting to tiny fluctuations.

Think of it like this:

- **MACD Line** = raw momentum (can be jumpy)
- **Signal Line** = calm, filtered version of MACD Line

Using 9 makes the signal line **responsive but still smooth**.

Analogy

Imagine you're listening to someone speaking:

- **MACD Line** = the person talking fast
- **Signal Line (9 EMA)** = a translator summarizing what they said clearly

The translator waits a bit before speaking, giving you a **cleaner version** of what’s happening.

The number **9** is like giving the translator **9 seconds** to process before responding.

🧪 Why not 5 or 15 instead of 9?

If the signal line used a shorter period like **5 EMA** of MACD:

- It would react too fast → more fake signals

If it used a longer period like **15 EMA**:

- It would react too slow → you miss entries

9 gives the best balance of:

- responsiveness
- smoothness
- reliability

That's why traders still use **12, 26, 9** 50+ years later.

In Short

- **12** EMA = short-term trend
- **26** EMA = medium-term trend
- **MACD Line = 12 EMA – 26 EMA**
- **Signal Line = 9-EMA of the MACD Line**

  The **9** is used to smooth MACD so buy/sell signals are more accurate.

EMA vs MACD vs RSI (Full Comparison)

1. What They Are

Indicator	Category	What It Measures
EMA	Trend Indicator	Trend direction (uptrend/downtrend)
MACD	Trend + Momentum Indicator	Trend strength, momentum, and reversals
RSI	Momentum Oscillator	Overbought/oversold conditions

2. Formula Difference

Indicator	Formula Style	# of EMAs Used
EMA	Weighted moving average formula	**1 EMA**
MACD	Difference of 2 EMAs + 1 EMA smoothing	**3 EMAs** (12, 26, 9)
RSI	Based on average gains vs losses	**No EMA directly**, but uses smoothing

3. What Each One Tells You

Indicator	Answers This Question
EMA	"Which direction is the price trending?"
MACD	"How strong is the trend? Is momentum increasing or decreasing?"
RSI	"Is price overbought or oversold? Is a reversal likely?"

4. Time Sensitivity (Speed of Signal)

Indicator	Speed
8/12 EMA	Fast
MACD	Medium
RSI	Medium-Fast
200 EMA	Slow

5. Best Use Case

Indicator	Best For
EMA	Determining trend and filtering trades
MACD	Timing entries/exits & spotting momentum shifts early

Indicator	Best For
RSI	Avoiding buying too high or selling too low

📊 Signals Provided

Indicator	Signal Type	Example
EMA	Crossover & trend	8 EMA crosses above 21 EMA → Buy
MACD	Momentum crossover & divergence	MACD crosses above signal line → Buy
RSI	Overbought/Oversold levels	RSI < 30 = oversold → possible buy

Analogy to Lock It in Your Brain

Imagine trading like **driving a car**:

Indicator	Analogy
EMA	Road direction: Are you going North (uptrend) or South (downtrend)?
MACD	Car engine & speed: Are you accelerating or slowing? Is engine losing power?
RSI	Fuel or heat gauge: Are you overheating (overbought) or running low (oversold)?

Put together:

- EMA tells you direction
- MACD tells you strength of the move
- RSI warns if you’re pushing too hard and a pullback is likely

Why Traders Use All 3 Together

Condition	Meaning
EMA Up	Trend direction is bullish

Condition	Meaning
MACD Up	Momentum supports the trend
RSI Not Overbought	You still have room to move higher

This combination makes a **strong buy zone**.

If any one disagrees, caution is needed.

Quick Memory Cheat Code

EMA = Trend
MACD = Momentum + Strength
RSI = Market Temperature

- Trend tells you **where to go**
- Momentum tells you **if the move is strong**
- RSI tells you **when to avoid jumping late**

Simple Example

Imagine BTC rising:

Indicator	Reading	Meaning
EMA above 200	Trend UP	Good environment to buy
MACD > Signal & rising	Strong momentum	Good timing
RSI = 65	Still below overbought	Safe to enter

But if RSI was 80, you avoid entering even if trend is up.

🚦 5. signal Column: Trade Direction

This column is created after combining EMA, RSI, and MACD conditions.

Example logic:

Condition	Signal
All bullish conditions met	+1 = Buy
All bearish conditions met	−1 = Sell
No valid signal	0 or NaN

This is the actual **trade decision** indicator.

6. position Column: Marks Entries & Exits

This calculates the change in the signal column from one candle to the next.

Example:

Previous Signal	Current Signal	Position Output
0 ⟶ +1	+1	Entry long position
+1 ⟶ 0	0	Exit long position
0 ⟶ −1	−1	Entry short position
−1 ⟶ 0	0	Exit short position

Note: The position column shows actual trade exposure; signal column only shows new trade trigger.

This tells your bot **when to open or close a trade**.

> In backtesting, position is crucial because it avoids repeatedly buying while already in a trade.

Why Combine All These Indicators?

One indicator alone is weak. Combining them creates a **multi-layer confirmation system**:

Indicator	Role
200 EMA	Trend filter, avoid wrong-side trades
8/21 EMA Cross	Momentum entry trigger
RSI	Momentum strength filter
MACD	Confirmation filter to avoid fake crosses
Signal & Position	Converts indicator data into trade execution logic

This structure helps build a **robust, low-noise strategy**.

A compact mapping of files and what they do:

- **`data/data_collector.py`**
  - Connects to Bitget (via `ccxt`) and fetches OHLCV candles.
  - Exposes `init_bitget(...)` and `fetch_ohlcv(...)`.
  - Returns clean `pandas.DataFrame` with columns `['timestamp','open','high','low','close','volume']`.
- **`strategies/strategies.py`**
  - Implements `ema_rsi_macd(df, ...)`.
  - Adds columns: `EMA_long_term` (200 EMA), `EMA50`, `EMA_short` (8 EMA), `EMA_long` (21 EMA), `RSI`, `MACD`, `MACD_signal`, `signal`, `position`.
  - `signal` is market-condition; `position = signal.diff()` marks discrete entry/exit on a bar.
- **`risk/risk.py`**
  - `position_size(balance, price, atr, ...)`: ATR-based sizing capped by a margin fraction.

  - `sl_tp_levels(entry_price, atr, ...)`: returns `(stop_loss, take_profit)` using ATR multipliers.
- **`paper_trader/paper_bot.py`**
  - `PaperBroker` class simulates leveraged futures-style trades (margin accounting, fees, cooldown after losing trades).
  - Methods: `open_long(...)`, `open_short(...)`, `close_position(...)`, `check_sl_tp(high, low)`.
- **`backtester/backtester.py` (a backtest utility)**
  - `backtest(df, initial_balance)`: simplified backtest for a DataFrame with `position` and ATR.
  - `compute_metrics(equity)`: basic performance metrics (total return, approximate annualization, Sharpe-like metric, max drawdown).
- **`alerts/telegram_alerts.py`**
  - Wrapper around `telebot` for Telegram alerts and CSV logging via `log_trade(...)`.
  - Implements per-chat rate limiting and message truncation to avoid Telegram errors.
- **`config.py`**
  - Loads credentials and settings from environment variables via `python-dotenv`.
- **`main.py`**
  - Orchestrates everything: fetches markets, computes indicators, runs `PaperBroker` simulation per timeframe/pair, calls `log_trade`, `send_alert`, prints final summaries and equity metrics.

Dataflow (end-to-end)

1. **Initialize exchange**
   `main.py` calls `init_bitget(api_key, secret, password, testnet=True)` to create a `ccxt.bitget` exchange instance (with sandbox/testnet option available).
   `exchange.options["defaultType"] = "swap"` configures the exchange client for futures/swap markets.

2. **Fetch candles**
   `main.py` calls `fetch_ohlcv(exchange, SYMBOL, timeframe, limit)` which returns a `pandas.DataFrame` with `timestamp, open, high, low, close, volume`. The function wraps `exchange.fetch_ohlcv` and converts types.

3. **Compute indicators & signals**
   `main.py` calls `ema_rsi_macd(df)` which:

   - computes `EMA_long_term` (200), `EMA50 for future strategy enhancements (currently not used for entry/exit logic)`, `EMA_short` (8), `EMA_long` (21),
   - computes `RSI` (14), `MACD` & `MACD_signal` (12,26,9),
   - sets `signal` column: `1` for bullish conditions, `-1` for bearish, `0` otherwise,
   - sets `position = signal.diff().fillna(0).astype(int)` marking the exact bars that open/close positions.

4. **Compute ATR**
   `main.py` computes `df['ATR'] =` AverageTrueRange(...).average_true_range() then uses .ffill().fillna(0.0) to carry ATR values forward.
   If the ATR value is **0 or missing**, the code applies a safety floor using:

atr = max(1.0, price * 0.001)
(The floor is only used when ATR = 0.0, not on every candle.)

5. **Construct PaperBroker**
   For each symbol/timeframe, `main.py` creates `broker = PaperBroker(initial_balance=INITIAL_BALANCE, leverage=LEVERAGE)`.

6. **Iterate through candles**
   For each `row` in `df.iterrows()` the flow is:

   - `ev = broker.check_sl_tp(row['high'], row['low'])`: checks intra-candle for SL/TP hits and returns a close event if triggered.
     - If SL/TP triggered: append `ev['balance']` to the equity curve, `log_trade(ev)` to CSV, and `send_alert(...)` to Telegram.
   - Read `signal = int(row.get('signal', 0))`.
     - If `signal == 1` and `broker.position == 0`: `ev = broker.open_long(price, atr, risk_pct)`: log and alert.
     - If `signal == -1` and `broker.position == 0`: `ev = broker.open_short(...)`.
     - If `signal == 0` and `broker.position != 0`: `ev = broker.close_position(price)`.
   - After every trade event (SL/TP, opening, or closing a trade), the bot updates the equity curve and calculates Total Return %, Win Rate %, and Max Drawdown %, which are logged and included in the Telegram alert instantly.

7. **Final summary**
   After the candle loop, **`main.py`** prints and sends a short final

summary per symbol/timeframe: final balance and remaining position.

That pipeline is:
Data → Indicators → Signals → Trade Simulation → Logging / Alerts → Summary & Metrics

Strategy Logic: Think of it like Driving with a Map

Imagine you're driving a car:

- **Macro trend (200 EMA):** This is like the main highway. If the highway is going north, you don't want to drive south, that's the long-term trend.
- **Momentum trigger (8 EMA vs 21 EMA):** This is like checking your speedometer and traffic lights. If the short-term speed is increasing above the medium speed, you know it's a green light to move forward.
- **Filters (RSI & MACD):** This is like checking the weather or traffic conditions. Even if the green light is on, you don't want to speed if the road is slippery (RSI too high) or the traffic is jammed (MACD not confirming momentum).

Buy Rule (Go forward):

1. Price is above the highway (200 EMA) → you're on the right road.
2. Short speed (8 EMA) is above medium speed (21 EMA) → car is accelerating.
3. RSI < 65 → not "too hot" to drive safely.
4. MACD confirms momentum → traffic conditions are good.

Sell Rule (Slow down / reverse):

1. Price is below the highway ⟶ wrong direction.
2. Short speed < medium speed ⟶ slowing down.
3. RSI > 35 ⟶ road not too empty (avoids overselling).
4. MACD confirms bearish momentum ⟶ traffic confirming slowdown.

Position Column (Signals for Action):

- The bot watches for changes in signals:
  - `0 → 1` = start moving forward (buy/long).
  - `1 → 0` or `1 → -1` = stop or reverse (exit long / enter short).
  - Same logic for short positions.

Why this combo works:

- Highway (200 EMA) prevents fighting the long-term trend.
- Speed checks (EMA crossovers) catch momentum quickly.
- Weather check (RSI & MACD) avoids bad trades.
- ATR acts like a suspension system, adjusting your risk according to road bumps (volatility).

2. ATR & Position Sizing: Think of it like Packing Your Backpack

- **ATR (Average True Range):** Measures how bumpy the road is.
  - TR = how far the car can swing left or right (high-low range).
  - ATR = average of these swings ⟶ tells you how "bumpy" the market is.
- **Position Sizing:** Decides **how much weight to put in your backpack** so you don't get hurt if you fall.

Steps:

1. **Risk per trade:** How much money you are willing to risk (like packing only what you can carry).
2. **Risk per unit:** How risky each item is (ATR × stop distance).
3. **Units by risk:** How many items you can carry safely (risk_amount / risk_per_unit).
4. **Max units by margin:** Don't overpack beyond your bag limit (margin-based cap).
5. **Final units:** The smaller of the two → ensures safety.

Stop Loss / Take Profit (SL/TP):

- Like setting a soft cushion for when you trip:
  - Stop-loss = how far you fall before exiting (ATR × multiplier).
  - Take-profit = how far you can safely run to collect rewards.

3. PaperBroker: The Simulated Playground

- **Think of it as Monopoly money:** The bot simulates real trading using pretend money.
- **Margin & Leverage:**
  - Deducts required money from your "wallet" when you open a trade.
  - Returns it when you close.
- **Dynamic SL/TP:** Adjusts based on bumpy roads (ATR vs price).
- **Cooldown:** If you "fall" (lose money), the bot waits 2 turns before trying again — avoids panic trading.
- **Perfect fills assumption:** The bot assumes trades happen exactly at the prices it wants — like perfect Monopoly moves.

4. Backtester Utility: The Practice Track

- Think of this as a **driving simulator**. You can test how the bot drives on past roads (historical data).
- Tracks total distance (total return), safe driving % (win rate), and worst accident (max drawdown).
- Quick checks before using the real playground (PaperBroker).

5. Alerts & Logging: The Assistant Who Watches You

- Sends messages to your phone (Telegram) for every trade.
- Logs all actions in a spreadsheet (CSV) so you can review later.
- Rate-limited to avoid spamming ⟶ doesn't shout at you every second.

6. What the Bot Can Do (v1)

- Follow trends safely.
- Open/close trades with momentum confirmation.
- Adjust trade size according to market bumps.
- Send real-time alerts and logs.
- Paper-trade safely with leverage and margin simulation.

What it Cannot Do (v1)

- Live trading without adjustments.
- Handle unexpected slippage or market chaos.
- Multi-asset risk management.
- Real partial fills, liquidity checks, or liquidation events.

Summary

Think of this bot as **a cautious, smart driver with a GPS**:

- GPS = EMA200 (long-term trend).
- Speed checks = EMA crossovers.
- Weather alerts = RSI & MACD.
- Safety gear = ATR + position sizing.
- Practice tracks = backtester & PaperBroker.
- Assistant = Telegram alerts & logs.

It won't take you over dangerous cliffs (risk control) and teaches you how to drive safely on bumpy roads (volatile markets).

Future Improvements for Version 2

1. Smarter Signal Logic

- **Current:** Uses EMA crossover, RSI, and MACD. Works well for trending markets but struggles in sideways/choppy markets.
- **Improvements:**
  - Add **multi-timeframe analysis**: Confirm signals on both lower and higher timeframes (e.g., 1h + 4h) to reduce false entries.
  - Introduce **volatility filters**: Only trade when ATR is above a minimum threshold to avoid low-volatility whipsaws.
  - Add **trend strength indicators**: Like ADX, to know if the trend is strong or weak before entering.

2. Better Risk Management

- **Current:** Uses fixed % risk per trade with dynamic ATR-based sizing.
- **Improvements:**
  - Add **dynamic risk allocation**: Reduce risk during drawdowns and increase during winning streaks (like Kelly Criterion or equity-based scaling).
  - Implement **max daily loss / max drawdown caps**: Stop trading for the day if losses exceed a threshold.
  - Multi-asset risk management: Allow the bot to trade multiple pairs while keeping total risk under control.

3. Realistic Trade Execution

- **Current:** Perfect fills assumed, no slippage, no partial fills.
- **Improvements:**

  - Simulate **slippage**: Randomly adjust fills based on recent market spreads.
  - Handle **partial fills** for big orders in low-liquidity markets.
  - Consider **order book depth** for better realism.

4. Advanced Position Management

- **Current:** Simple ATR-based SL/TP and cooldowns.
- **Improvements:**
  - **Trailing stop-loss:** Lock in profits as the price moves favorably.
  - **Partial exits:** Close part of the position at certain TP levels to secure gains.
  - **Dynamic TP scaling:** Increase TP multiplier based on trend strength or volatility.

5. Alerts & Monitoring Enhancements

- **Current:** Telegram alerts for trades, basic logging.
- **Improvements:**
  - **Customizable alert types:** Notify only for big wins/losses or missed opportunities.
  - **Web dashboard:** Real-time PnL, open positions, and equity curves in a web interface.
  - **Email/SMS alerts** for backup notifications.

6. Performance & Backtesting

- **Current:** Backtester is simple; PaperBroker simulates trades.
- **Improvements:**
    - **Monte Carlo simulations:** Test robustness by simulating trades with randomized sequences.
    - **Walk-forward analysis:** Validate strategy on unseen future data after backtesting.
    - **Sharpe, Sortino, and drawdown ratios** improvements: More accurate risk-adjusted metrics.

7. Machine Learning / Adaptive Strategies

- Introduce **adaptive EMA lengths** based on market conditions.
- Predict short-term volatility spikes using historical ATR patterns.
- Use reinforcement learning to optimize position sizing and SL/TP dynamically.

8. Safety & Fail-safes

- **Current:** Basic margin checks.
- **Improvements:**
    - **Auto stop-loss for catastrophic events** (e.g., exchange outages, flash crashes).
    - **Connectivity monitoring:** Pause trading if API responses are delayed or failing.

    - **Version control / rollback:** Save checkpoints to revert the bot to previous stable logic.

Version 2 of the bot could:

- Trade **more reliably** in sideways and volatile markets.
- Protect capital better with **smarter risk management**.
- Simulate **realistic market conditions** including slippage, partial fills, and liquidity constraints.
- Offer **better monitoring & reporting**, including dashboards and alerts.
- Gradually become **adaptive and predictive**, learning from past market behavior.

Essentially, V2 would be a **smarter, safer, and more professional trading companion.**