

Project Overview: Algorithmic Trading Bot for Indian Markets

BY: Aryan Gupta

This project involved designing and developing a fully automated algorithmic trading system capable of running multiple trading strategies concurrently in the Indian stock and options market using live market data. The system autonomously processes live ticks, evaluates strategy logic, manages risk, and executes trades through ICICI's Breeze API.

System Architecture

1. Core Execution Layer

The system is orchestrated from a central script — `run_trading.py` — which serves as the entry point for the trading engine.

- On initialization, it:
 - Establishes a live data stream from ICICI Breeze WebSocket for OHLCV tick data.
 - Launches multiple strategy threads, each independently processing live data for assigned symbols.
 - Starts an RMS (Risk Management System) thread to monitor and control overall portfolio risk.
-

2. Strategy Engine

The system currently runs 7 strategies concurrently:

- 5 core strategies:
 - 4 equity-based strategies using technical indicators and trend-following logic.
 - 1 options strategy based on momentum and volume signals.
- 2 supplementary strategies:
 - EMA Crossover Strategy – uses short-term and long-term exponential moving averages to detect momentum shifts.
 - Simple Strategy – enters a long position when `close > previous_close` and exits when `close < previous_close`.

Each strategy runs in parallel threads, one per (`symbol × strategy`) pair, enabling concurrent evaluation of multiple instruments.

3. Data Flow

Live Tick Stream → Strategy Evaluation → Signal Generation → RMS Approval → Order Execution

1. Live Tick Stream

The Breeze API streams real-time OHLCV data for subscribed symbols.

Each incoming tick is broadcast to all strategy threads via a thread-safe queue.

2. Strategy Evaluation

Each strategy consumes the tick and updates its internal state (moving averages, indicators, etc.).

If trade conditions are met, the strategy generates a buy/sell signal.

3. Pre-Trade RMS Check

Before execution, every trade passes through the pre-trade RMS layer:

- Checks symbol-level and portfolio-level exposure.
- Ensures risk limits and VaR (Value at Risk) thresholds are not breached.
- If the trade is within limits, RMS approves execution.

4. Order Queue & Execution

Approved trades are added to the order queue, and an order is placed through the Breeze API at the latest market price.

5. Trade Lifecycle Management

The system continuously monitors each open position and exits when the opposite signal or stop condition is triggered.

4. Risk Management System (RMS)

The RMS thread continuously monitors:

- Portfolio exposure across all running strategies.
 - Per-symbol exposure and margin usage.
 - Value at Risk (VaR) and overall capital allocation.
It dynamically prevents over-leveraging and excessive drawdowns by restricting trade approvals when thresholds are breached.
-

5. Data Persistence and Logging

- Trade Database:

All executed trades, positions, and closed trade logs are stored in a SQLite database (trades.db).

This enables complete state persistence, performance tracking, and post-trade analytics.

- System Logs:

Every system event, signal, RMS decision, and order action is logged in focal_quant.log, providing detailed traceability and debugging insights.

6. Concurrency and Scalability

- Each (strategy × symbol) combination runs in its own thread, enabling true parallelism.
 - The architecture supports scaling to more instruments or strategies simply by spawning new threads.
 - RMS and data streams run independently to maintain responsiveness even under heavy load.
-

7. Continuous Operation in Live Market

The system operates in live market conditions, processing:

- Real-time price updates
- Dynamic trade execution
- Live risk monitoring

It is designed for robust fault tolerance, with error handling for data stream interruptions, order rejections, and database locks.

Backtesting Framework & Strategy Evaluation

To ensure the robustness and profitability of the trading algorithms before live deployment, I developed a comprehensive backtesting pipeline using the Backtrader framework.

All five strategies were tested on 6 months of historical market data (1-minute resolution), covering diverse market conditions and volatility phases.

Backtesting Setup

- Framework: Backtrader
- Data Range: 6 months of historical data for each traded symbol
- Assets Covered: 4 equity instruments and 1 options contract
- Data Source: Historical OHLCV data fetched from ICICI Breeze API and stored locally
- Storage Format: Each backtest result is saved as
- {strategy_name}_05-11_results.csv

in the project root directory, containing detailed performance metrics.

ADX Power Momentum Strategy

Core Concept

This strategy looks for “power moves” — situations where the trend is strong (high ADX) and the direction (bullish or bearish) is clearly dominant.

It confirms this with the relationship between +DI, -DI, and ADX, then validates with short-term and medium-term EMA alignment, ensuring momentum is in the same direction as the trend.

Once a valid trend is found, it enters the trade and uses volatility-based stop-loss (2×ATR) and take-profit (4×ATR) levels to manage the position.

⌚ Step 1. Indicators and Setup

The system uses six main indicators:

ADX (Average Directional Index)

Measures trend strength (but not direction).

- High ADX (above 30) = strong trend
- Low ADX (< 20) = sideways market

+DI (Positive Directional Indicator)

Measures the bullish directional movement.

When +DI is above both ADX and -DI, it means buyers are in control.

-DI (Negative Directional Indicator)

Measures the bearish directional movement.

When -DI dominates, sellers control the market.

ATR (Average True Range)

Measures volatility — how much the market typically moves per bar.

Used to scale stop-loss and take-profit dynamically.

EMA(5)

Short-term trend indicator → tracks immediate momentum.

EMA(14)

Medium-term trend indicator → tracks broader price direction.

⌚ Step 2. The Philosophy

It combines trend confirmation (ADX/DMI) with momentum confirmation (EMA crossover).

In plain words:

**“Only buy when the trend is strong and short-term momentum is accelerating upward.
Only sell when the trend is strong and momentum is accelerating downward.”**

This prevents entries during choppy or undecided periods.

Step 3. Long (Buy) Setup

For a long (buy) trade, all of the following must align:

1. ADX above 30 → confirms a strong trend environment.
2. +DI above ADX and -DI → buyers are leading.
3. ADX - -DI is at least 5× larger than +DI - ADX → strong bullish imbalance.
4. EMA(5) > EMA(14) → short-term price is above medium-term trend → bullish momentum confirmed.

When these align, the strategy places a limit buy order at the current close price.

Step 4. Short (Sell) Setup (currently commented out)

The short setup mirrors the long one, just in reverse:

1. ADX > 30 → strong trend.
2. +DI below ADX, ADX below -DI → sellers dominate.
3. |+DI - ADX| ≥ 5×|ADX - -DI| → bearish imbalance.
4. EMA(5) < EMA(14) → short-term trend is down.

If this were active, it would place a limit sell order to enter short.

Step 5. Trade Execution and Risk Management

When a trade executes (i.e., the order is filled), the system immediately sets:

- Take Profit (TP) = Entry Price ± (4 × ATR)
- Stop Loss (SL) = Entry Price ± (2 × ATR)

So, the Risk : Reward ratio = 1 : 2

— a balanced structure that targets large gains relative to controlled risk.

For example:

- If volatility (ATR) = ₹10 and the entry is at ₹100,
 - SL = ₹80
 - TP = ₹140
- Aiming to capture a “power move” while limiting downside.

Step 6. Trade Management (in next())

Once a trade is open:

- If price hits the take profit level, it closes the trade with a limit sell (for long) or limit buy (for short).
- If price hits the stop loss level, it immediately closes the trade with a market order.

So the strategy is fully self-contained: once a position is open, no new trades are entered until it's closed.

Step 7. Momentum and Trend Reinforcement

The EMA crossover ensures trades are in sync with short-term momentum.

This avoids entering on stale or fading moves.

- When $\text{EMA}(5) > \text{EMA}(14)$, the short-term momentum is rising faster than the mid-term average → bullish pressure.
- When $\text{EMA}(5) < \text{EMA}(14)$, momentum is fading → bearish environment.

This acts as a *trend filter* that aligns entries with the broader market flow.

OPEN RANGE BREAKOUT STRATEGY

Core Idea

This strategy is built on the Open Range Breakout concept — the idea that the market's direction for the day is often determined by whether price breaks above or below the range formed during the first few candles of the session.

It combines this time-based breakout logic with:

- Trend confirmation using ADX and DMI,
 - Momentum validation using VWAP and moving averages,
 - ATR-based risk management,
to improve reliability and filter out false breakouts.
-

Step 1. Identify the “Opening Range”

At the start of each trading day, the strategy records the high and low prices of the first 30 bars (this number could represent, for example, the first 30 minutes or first 6 candles depending on your data).

This forms the “opening range” — essentially the first balance zone of the day.

- The highest high during this period is called the *opening range high*.
- The lowest low is called the *opening range low*.

These two prices act as breakout levels for the rest of the day.

Step 2. Wait for the Breakout

After the first 30 bars, the strategy waits to see if the current price breaks out above the opening range high (potential bullish breakout) or below the opening range low (potential bearish breakout).

However, it doesn't just react to any breakout — it demands additional trend and momentum confirmations before taking a position.

Step 3. Confirm with Trend Strength Indicators

To ensure that the breakout is supported by genuine market momentum, several technical filters are used:

ADX (Average Directional Index)

- This measures how strong the current trend is.
- The strategy only considers trades if the ADX is above 30, meaning the trend strength is significant.

2 DMI (+DI and -DI)

- These two lines show whether the bulls or bears are in control.
- For a long trade, the bullish +DI must be higher than both ADX and -DI (showing buyers are dominating).
- For a short trade, the opposite must be true — the bearish -DI must dominate.

3 ADX Difference Filters

- The strategy measures the distance between ADX and DMI lines.
 - It checks that one direction's strength (e.g., +DI for long) is much larger than the opposite direction's.
 - This helps filter out situations where the trend is unclear or the two sides are balanced.
-

4 Step 4. Confirm with Momentum and Trend Filters

Even if the breakout and ADX/DMI look good, the strategy still validates momentum using other indicators:

1 VWAP (Volume Weighted Average Price)

- VWAP represents the average price weighted by traded volume.
- For longs, price must be above VWAP — confirming that buyers are pushing prices above the average traded level.
- For shorts, price should be below VWAP — confirming seller dominance.

2 Moving Averages (EMA and SMA)

- The strategy uses short-term exponential and simple moving averages (3-period and 5-period).
- For a long trade, the fast EMA should be above the SMA — confirming upward momentum.
- For a short trade, EMA below SMA confirms downward bias.

All of these together create two sets of “long conditions” and two sets of “short conditions” — one using VWAP, and another using EMA/SMA crossovers — to strengthen confirmation before entering a trade.

5 Step 5. Entry Logic

Once all the above conditions align:

- The price breaks above the opening range high,
- ADX is strong,

- DMI confirms direction,
- VWAP and EMA/SMA confirm upward momentum,

the strategy triggers a long entry.

It places a limit order at the current price to buy, meaning it enters at or near that breakout level.

The position size is automatically calculated based on the broker's total capital and the current price — so it uses nearly the full capital in one trade (1x leverage, for simplicity).

Step 6. Stop Loss and Take Profit Setup

After a long trade is executed:

- The stop loss (SL) is set below the lowest low of the past few candles (specifically the lowest price of the previous six bars).
This ensures the stop is placed just below recent structure, protecting from normal fluctuations.
- The risk per trade is measured as the difference between the entry price and this stop loss level.
- The take profit (TP) is then set to four times that risk distance above the entry price, giving a risk-reward ratio of 1:4.
So for every ₹1 risked, the potential gain is ₹4.

Step 7. Exit Logic (Position Management)

Once in a trade, the strategy continuously monitors:

- If price drops below the stop loss, it immediately exits the long position and logs the event as a "LONG SL HIT".
- If price reaches or exceeds the take-profit level, it exits and logs a "LONG TP HIT".

The same logic would apply for short positions (if they were activated), but in this version the short entries are commented out.

Step 8. Daily Reset Logic

At the start of a new day:

- The opening range high and low lists are cleared and recalculated based on the new day's first 30 bars.
- This ensures that each trading day has its own breakout levels, independent from the previous day.

Step 9. Optional Time-Based Exit (Commented Out)

There's also a time-based exit condition (commented in the code) which would close any open positions at 3:20 PM, to avoid holding trades into market close.

That's common in intraday systems — even if the target or stop loss isn't hit, all trades are closed before the end of the day to manage overnight risk.

Philosophy Behind It

This strategy combines the time-based breakout logic of an ORB with trend confirmation from ADX/DMI and momentum confirmation from VWAP and moving averages.

That means it doesn't just react to price crossing a line — it checks *whether the market has the strength and volume conviction* to sustain that move.

So effectively it says:

"I'll only trade a breakout if it's backed by strong directional momentum, expanding volatility, and institutional participation above VWAP."

This filters out 90% of fake breakouts and gives you a cleaner, higher-quality trade setup.

Strategy 3 — Conceptual Overview

This strategy attempts to:

- Enter when the trend strength (ADX) suddenly increases beyond its “normal” range.
- Confirm the direction using the +DI / -DI crossover and relative strength.
- Exit when the momentum weakens, or profit/stop-loss targets (based on ATR) are hit.

It's designed to catch volatility breakouts backed by statistically significant ADX changes.

Indicators Used

Indicator	Description	Role in Strategy
ADX (Average Directional Index)	Measures trend <i>strength</i> (not direction).	Detects strong trends.
+DI / -DI (Directional Movement Index)	Indicates <i>trend direction</i> .	Confirms whether bulls or bears are leading.
ATR (Average True Range)	Measures <i>volatility</i> .	Used to set dynamic SL and TP.
Rolling mean/std of ADX	Measures “normal” ADX behavior.	Detects when ADX breaks above normal volatility levels.
Rolling mean/std of (DMP – DMN)	Detects shifts in directional dominance.	Used as an <i>exit filter</i> .
SMA(5)	Short-term trend filter on price.	Used to confirm exit conditions.

DMI Difference Strength

This shows how much buying pressure dominates over selling pressure on average.

Used for exit conditions — when diff drops below its mean – std, trend is weakening.

Entry Logic

Inside next(), it checks when to enter a trade:

```
if (self.adx[-1] > (self.adx_mean[-1] + self.adx_std[-1])  
    and self.dmp[-1] > self.adx[-1]  
    and self.adx[-1] > self.dmn[-1]):
```

Breakdown:

Condition	Meaning
ADX > (mean + std)	ADX is breaking above normal → trend strength is <i>spiking</i>
+DI > ADX	Uptrend direction confirmed
ADX > -DI	ADX rising in favor of bullish direction

In other words —

“The trend strength just exploded, and the bullish direction is dominating.”

It then places a limit buy order at the current close:

```
limit_price = self.data.close[-1]  
self.order = self.buy(size=1, exectype=bt.Order.Limit, price=limit_price)
```

Stop Loss & Take Profit Setup

When the buy order executes:

```
self.entry_price = price  
self.tp = price + (self.atr[0] * 4)  
self.sl = price - (self.atr[0] * 2)
```

✓ Meaning:

- Take profit target = $+4 \times \text{ATR}$ above entry.
- Stop loss = $-2 \times \text{ATR}$ below entry.
- So Risk:Reward = 1:2, a healthy asymmetric setup.

Exit Logic

Once in a position, the strategy watches for two main exits:

(a) Take-Profit Exit

```
if (self.tp is not None and self.data.close[-1] >= self.tp  
    and (self.diff[-1] < (self.diff_mean[-1] - self.diff_std[-1]))  
    and self.data.close[-1] < self.sma_5[-1]):
```

Breakdown:

Condition	Meaning
price >= TP	Reached target
diff < mean - std	DMI momentum weakening
price < SMA(5)	Short-term reversal signal

→ So even if TP is hit, it exits *only if the market starts showing exhaustion* (diff weakening and price rolling over).

That avoids premature exits during strong trends.

(b) Stop-Loss Exit

```
elif (self.sl is not None and self.data.close[-1] <= self.sl  
    and (self.diff[-1] < (self.diff_mean[-1] - self.diff_std[-1]))):
```

Meaning:

- Price dropped below stop loss
- DMI difference showing weakening ($\text{diff} < \text{mean} - \text{std}$)

→ In short: “Trend reversed and buying power vanished — exit immediately.”

Practical Interpretation

This strategy is designed for trend breakout detection with volatility & statistical confirmation.

Component	Function
ADX spike	Confirms breakout strength
+DI dominance	Confirms trend direction
ATR	Defines risk levels dynamically
Mean + Std filters	Avoids false signals (<i>only reacts to statistically significant surges</i>)
Short-term SMA & diff filters	Confirms when momentum truly fades for exit

Example Walkthrough

Time	ADX	+DI	-DI	ATR	ADX_mean	ADX_std	diff	Signal	Action
09:30	28	35	10	3.0	22	4	25	ADX > mean+std (26), +DI > ADR	<input checked="" type="checkbox"/> Buy
10:30	32	40	9	3.1	-	-	31	Uptrend continues	Hold
11:15	36	32	15	3.2	-	-	17	diff ↓ below mean-std	<input type="triangle-down"/> Exit
11:30	35	28	20	3.1	-	-	8	Price < SMA(5)	<input checked="" type="checkbox"/> Sell / Exit

Parameters Summary

Parameter	Default Meaning
ADX period	14 Measures trend strength
ATR period	14 Volatility measure for SL/TP
ADX SMA window	20 Rolling mean for “normal” ADX
ADX StdDev window	20 Threshold for detecting ADX breakout
TP	4 × ATR Target profit
SL	2 × ATR Stop loss
SMA(5)	5 Short-term trend reversal check

Core Philosophy

It's based on a volatility expansion + trend confirmation model:

“Only enter when the trend’s strength (ADX) statistically spikes beyond normal levels and the bullish direction is clearly dominant.

Manage exits dynamically using volatility (ATR) and directional decay.”

This makes it ideal for momentum bursts — e.g., NIFTY / BANKNIFTY breakouts or post-news volatility.

ADX and SuperTrend Strategy

This strategy — called ADX-DMI-SupertrendStrategy — aims to catch strong directional trends early and manage them with dynamic stop loss (SuperTrend) and volatility-based take-profit levels (ATR).

Key Indicators Used

Indicator	Purpose	Explanation
ADX (Average Directional Index)	Measures <i>trend strength</i>	High ADX (>40) = strong trend
DMI (Directional Movement Index)	Identifies <i>trend direction</i>	+DI > -DI → uptrend, -DI > +DI → downtrend
ATR (Average True Range)	Measures <i>volatility</i>	Used to confirm volatility expansion and set SL/TP
ATR Mean (SMA of ATR)	Measures <i>baseline volatility</i>	Compare current ATR vs. average volatility
SuperTrend	Defines <i>dynamic trailing stop</i>	Uses $ATR \times \text{multiplier}$ to move SL dynamically

SuperTrend Indicator (Custom Defined)

```
class SuperTrend(bt.Indicator):
```

```
    lines = ('supertrend', 'upperband', 'lowerband')
```

```
    params = (('period', 7), ('multiplier', 1))
```

- It computes the upper and lower bands around the mid-price ($HL2 = (\text{high} + \text{low})/2$).
- Each band = $HL2 \pm (ATR \times \text{multiplier})$
- The supertrend line flips between the upper and lower band depending on where price closes.

Interpretation:

- When price closes above the previous upper band → bullish → supertrend = lower band.
- When price closes below → bearish → supertrend = upper band.

Essentially, it acts like a dynamic trailing stop following the trend.

Strategy Core Logic

→ So the strategy always knows:

- ADX strength
 - Whether +DI or -DI dominates
 - How volatile the market is right now
 - And where to place stop loss (from SuperTrend)
-

Entry Logic (Inside next())

Each bar, it checks if the market meets conditions for either a LONG or SHORT setup.

LONG (Buy) Setup:

```
long_condition = (  
    (adx > dmi_plus or dmi_plus > adx) and  
    adx > 40 and  
    atr > atr_mean  
)
```

Breakdown:

1. ADX > 40 → trend is strong
2. DMI+ > ADX (or vice versa) → uptrend direction confirmed
3. ATR > ATR_mean → volatility expansion (momentum confirmed)

If all true:

- The trend is strong and actively expanding upward.
-

SHORT (Sell) Setup:

```
short_condition = (  
    (adx > dmi_minus or dmi_minus > adx) and  
    adx > 40 and  
    atr > atr_mean  
)
```

Same logic, but for downtrend confirmation:

- DMI- dominates → downtrend

- ADX high + ATR > mean → volatility expanding → strong sell signal.
-

Order Execution

If not already in a position, it executes a bracket order:

```
self.order = self.buy_bracket(  
    price=self.entry_price,  
    stopprice=self.stop_loss,  
    limitprice=self.take_profit  
)
```

That's a three-in-one trade:

- Main entry (buy/sell)
- Automatic stop-loss
- Automatic take-profit

How SL & TP are computed:

For Long:

```
self.stop_loss = supertrend.lowerband[0]  
  
risk = entry_price - stop_loss  
  
take_profit = entry_price + (risk * risk_reward)
```

For Short:

```
self.stop_loss = supertrend.upperband[0]  
  
risk = stop_loss - entry_price  
  
take_profit = entry_price - (risk * risk_reward)
```

So your Reward : Risk ratio = risk_reward parameter (default 2:1)

Exit & Trade Management

The bracket order automatically handles exits:

- If price hits the stop_loss → exit trade with loss.
- If price hits the take_profit → exit trade with gain.

Additionally, each trade's result (PnL) is logged when it closes:

Strategy Filters

This strategy trades only during strong, trending, high-volatility phases:

Filter	Purpose
ADX > 40	Ensures trend strength
ATR > ATR_mean	Ensures volatility expansion
SuperTrend	Provides volatility-adjusted dynamic SL
risk_reward = 2.0	Maintains consistent R:R ratio

Together, these filters reduce false signals in choppy periods.

Typical Trade Behavior

Market Type	Action	Expected Outcome
Strong uptrend	Long entry when ADX & DMI+ rise	Capture sustained move
Strong downtrend	Short entry when ADX & DMI- dominate	Capture bearish leg
Low volatility / sideways	No trades (ADX low, ATR < mean)	Avoid noise

Example Trade Flow

Time	ADX	+DI	-DI	ATR	ATR_mean	Close	Signal	SL	TP
10:00	42	28	15	3.1	2.7	15800	BUY	15700	16000
10:45	46	35	12	3.5	2.8	15980	Hold	-	-
11:15	51	38	11	3.8	2.9	16010	TP hit	EXIT	-

Philosophy Behind the Strategy

It's a trend + volatility breakout model:

- Trades only strong, expanding trends.
 - Uses SuperTrend to anchor stop losses at logical, volatility-adjusted levels.
 - Sets take-profit automatically, enforcing consistent R:R control.
-

Option Price Momentum Strategy

Conceptual Overview

The **Option Price Momentum Strategy** is a **simple, data-driven short-term trend-following model** applied to **option prices** rather than underlying stocks or indices.

It uses **momentum in price, volume, and open interest (OI)** to identify moments when **option buyers are aggressively building new positions**, suggesting continuation in the same direction.

The Core Logic

Inputs — what data it needs:

From your call_df, each row includes:

- open, high, low, close — the option's OHLC prices
 - volume — number of contracts traded
 - oi — open interest (outstanding contracts, i.e., positions still open)
-

Indicators it computes:

For each candle (row), the strategy computes **Simple Moving Averages (SMA)** over the last sma_period (default = 20 bars) for:

Indicator Description

sma Average of closing prices → price trend baseline

vol_sma Average of traded volume → volume trend baseline

oi_sma Average of open interest → participation trend baseline

These moving averages give "normal" levels — so the current values can be compared to detect surges.

Entry (BUY) conditions:

It **enters a long position (buys)** when **all three** conditions align:

price_up = close > sma

vol_up = volume > vol_sma

oi_up = oi > oi_sma

Interpretation:

- **Price > SMA:** option price is breaking above its average → bullish momentum.
- **Volume > avg volume:** traders are actively trading more than usual → strong interest.

- **OI > avg OI:** new positions are being added (not just closing old ones) → conviction behind the move.

So, you only enter when there's **price strength + participation + conviction**.

In options terms: this typically happens when **the underlying is moving in favor of the option buyers**, e.g. for a **Call**, when the index is trending up.

Exit (SELL) conditions:

It **closes the long position** (sells) when either:

price < sma

OR

volume < vol_sma

Interpretation:

- Price falling back below average → momentum is fading.
- Volume drying up → participation slowing → trend losing steam.

It's a **momentum decay exit** — you ride the wave until participation weakens.

No shorting:

The strategy only goes **long → exit → flat**.

There's **no short position** or directional bias reversal — it only participates on bullish momentum bursts.

💡 Example Scenario (for a Call Option)

Date	Close	Volume	OI	SMA(20)	vol_SMA	oi_SMA	Action
Day 20	120	1500	9500	119	1200	9100	BUY (price breakout + volume surge + OI rising)
Day 23	124	1600	9700	121	1300	9200	Hold
Day 27	118	1100	9400	120	1280	9300	EXIT (price < SMA and volume down)

Result → PnL = (Exit - Entry) × Position size

💡 Risk & Trade Behavior

Aspect	Behavior
Holding period	Short (depends on how long price stays above SMA)
Stop loss	Implicit — exits when volume or price weakens
Reward	Captures strong upward moves in the option premium
Best in	Trending underlying (e.g. NIFTY or BANKNIFTY trending sharply)
Weak in	Choppy or sideways market (frequent whipsaws)

Why It Works (sometimes)

- Price > SMA = momentum confirmation
- Volume > vol_SMA = real buying activity (not just price noise)
- OI > oi_SMA = new positions are added (fresh money entering)

This 3-way confirmation filters out many false signals compared to just using price.

It's like saying:

"Don't just show me a rising price — show me that more traders are joining the move and keeping their positions open."

Implementation Notes

- **Backtrader** framework runs it bar-by-bar (like real trading).
- Each next() call checks if an entry/exit signal exists.
- When a trade closes, PnL is recorded and reported.
- The full backtest includes analyzers: Sharpe, SQN, Drawdown, CAGR, etc.

Backtest Results:

UNDERLYING: NIFTY25600CE

Expiry: 2025-09-16

TradeAnalyzer: Trades=202 Wins=116 Losses=86 NetPnL= **-54.49**

DrawDown: Max DD=0.06% Duration=2966

CAGR: -1.70% | Total Return: -0.05%

SQN: -2.35