How I Built an AI-Powered Trading Bot That Achieved a 70% Win Rate

Raditya Follow 8 min read · Jul 31, 2025

A deep dive into the data, the walk-forward methodology, and the Smart Money Concepts that power my XAUUSD trading algorithm.

The Elusive Goal: From Chasing Indicators to Building a System

Building a consistently profitable trading bot is the holy grail for many algorithmic traders. My journey, like many others, began with a frustrating cycle of chasing "magic indicators" and testing strategies that looked incredible on paper but crumbled under the pressure of live market data.

This isn't a story about finding that one magic indicator. It's a detailed breakdown of a systematic and disciplined approach that finally bore fruit. After an extensive development process, I created a Python-based bot for XAUUSD (Gold) that achieved a **70% win rate** and a **5,381% gross return** in a highly realistic, multi-year backtest designed to simulate the unforgiving conditions of a proprietary trading firm.

This is how I did it, pillar by pillar.

Pillar 1: The Foundation — High-Quality Historical Data

Any algorithm is only as good as the data it learns from. For this project, I needed a deep, clean, and reliable historical dataset across multiple timeframes. The data for XAUUSD — spanning M30, H1, H4, and D1 — was sourced from a reputable provider known for extensive historical data.

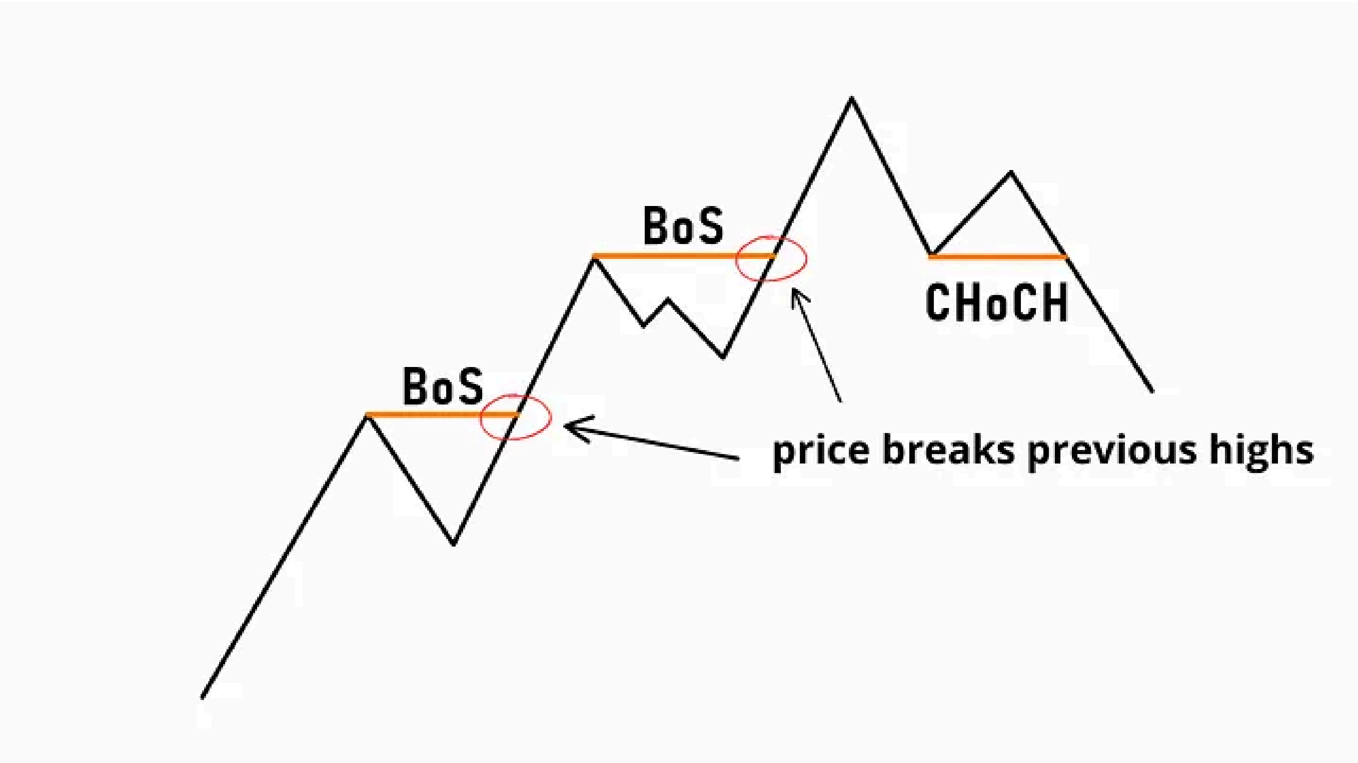
> skyscraper on sand. It guarantees failure. Investing in good data was the non-negotiable first step.

Pillar 2: The Strategy — Why Smart Money Concepts (SMC)?

Instead of relying on lagging indicators like RSI or Moving Averages, I chose Smart Money Concepts (SMC) as the strategic core. SMC provides a logical framework for reading price action by tracking the potential footprints of institutional players.

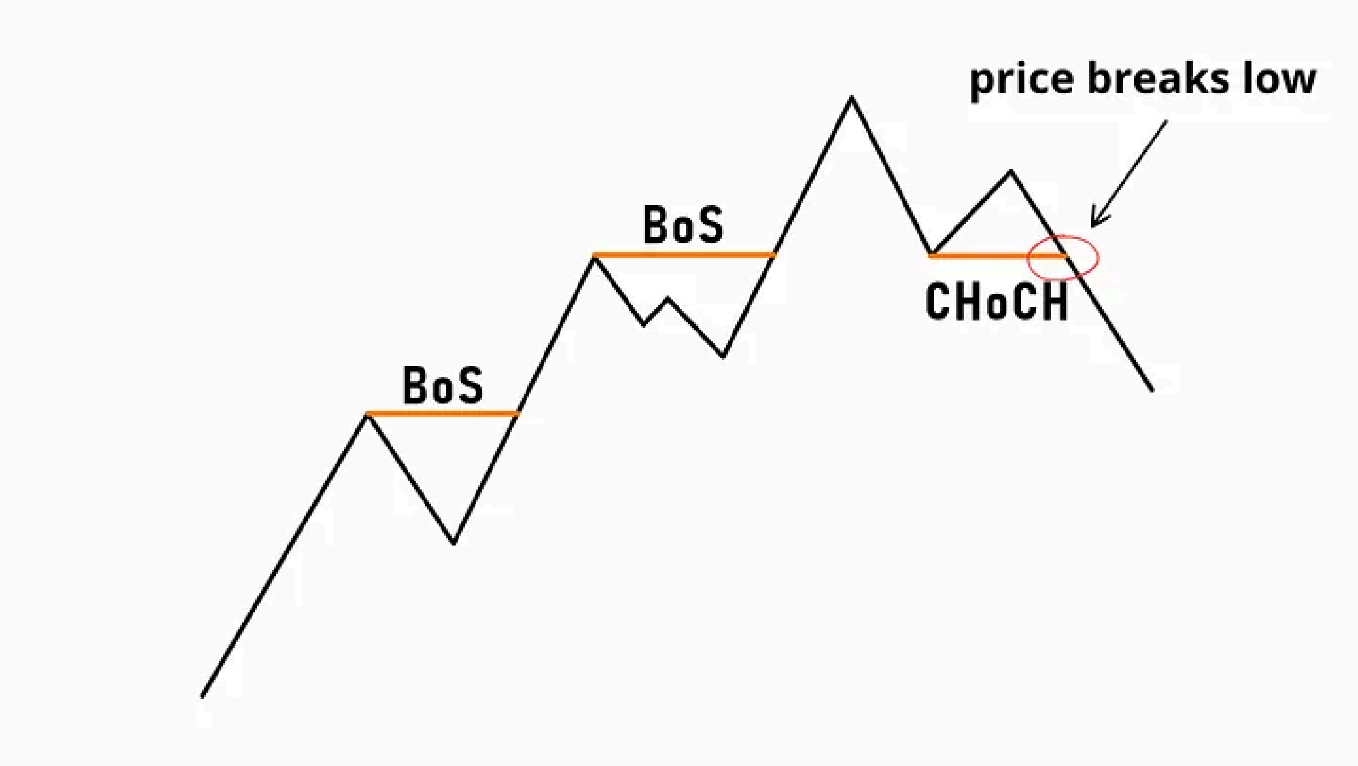
My bot's logic is built on a strict, four-step sequence that must be confirmed before a trade is even considered.

1. Establishing Directional Bias (H4)



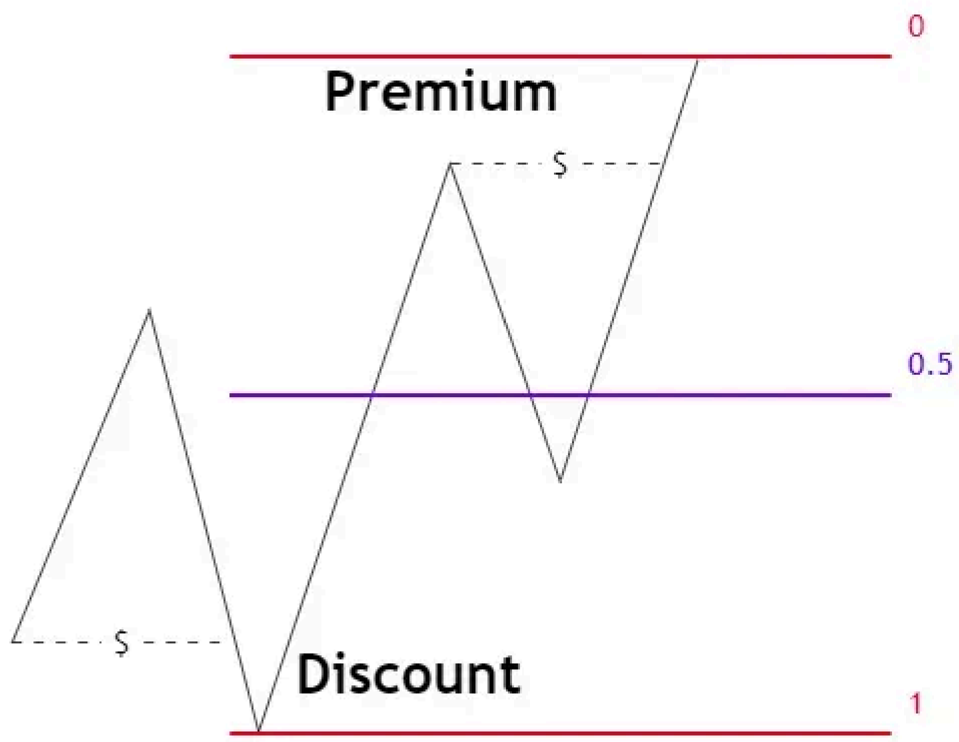
BoS Structure

First, the bot analyzes the H4 chart to identify major swing points. It determines the overall market structure by identifying Breaks of Structure (BoS) and Changes of Character (CHoCH). This establishes the high-level directional bias. If the trend is bullish, it will only look for buys; if bearish, only sells.



CHoCH Structure

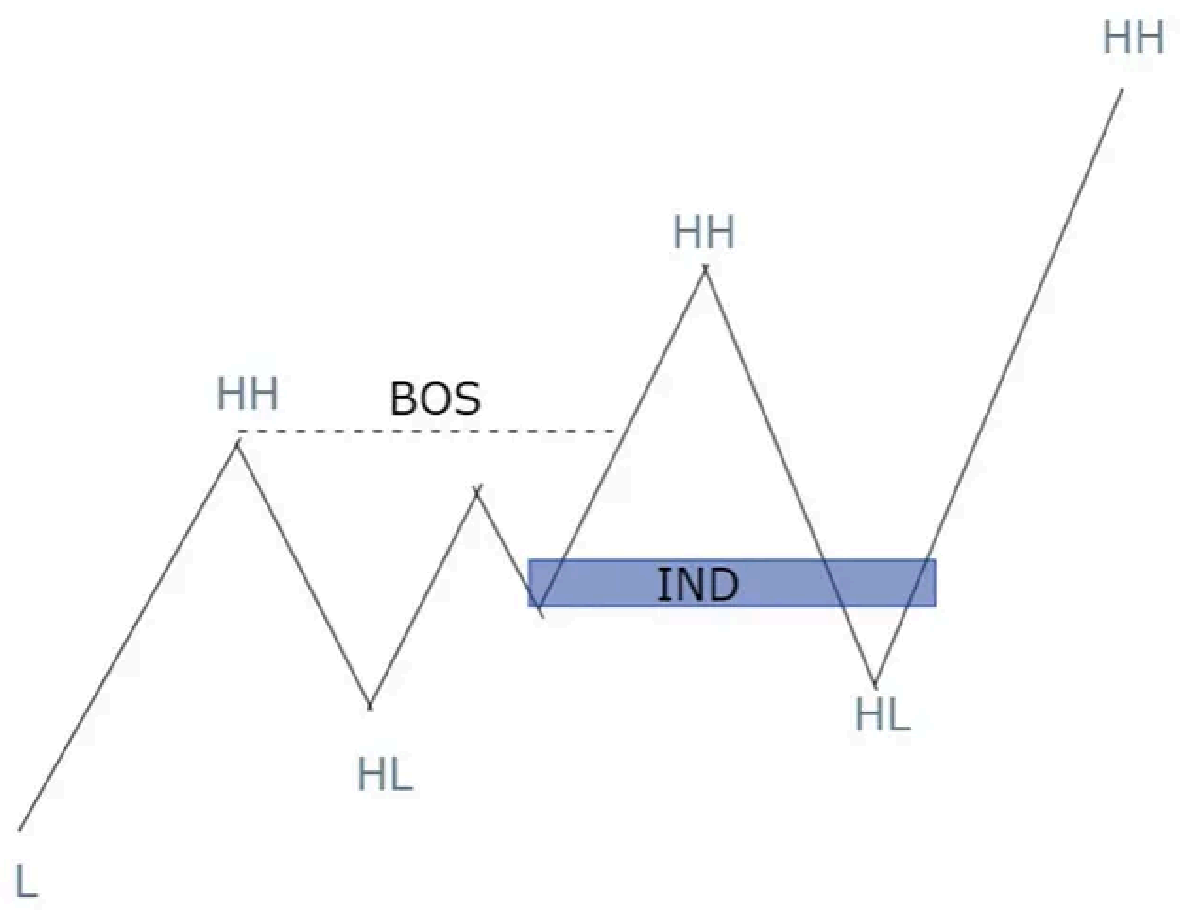
2. Trading from the Right Zones (H4)



Premium Discount Array (PDA)

Trading is about buying low and selling high. In a bullish H4 trend, the bot will **only** look for buy signals if the price has pulled back into a "discount" zone (below the 50% equilibrium level of the H4 range). Conversely, in a bearish trend, it **only** looks for sell signals in the "premium" zone (above the 50% level). This simple rule prevents the bot from chasing pumps or panic-selling dumps.

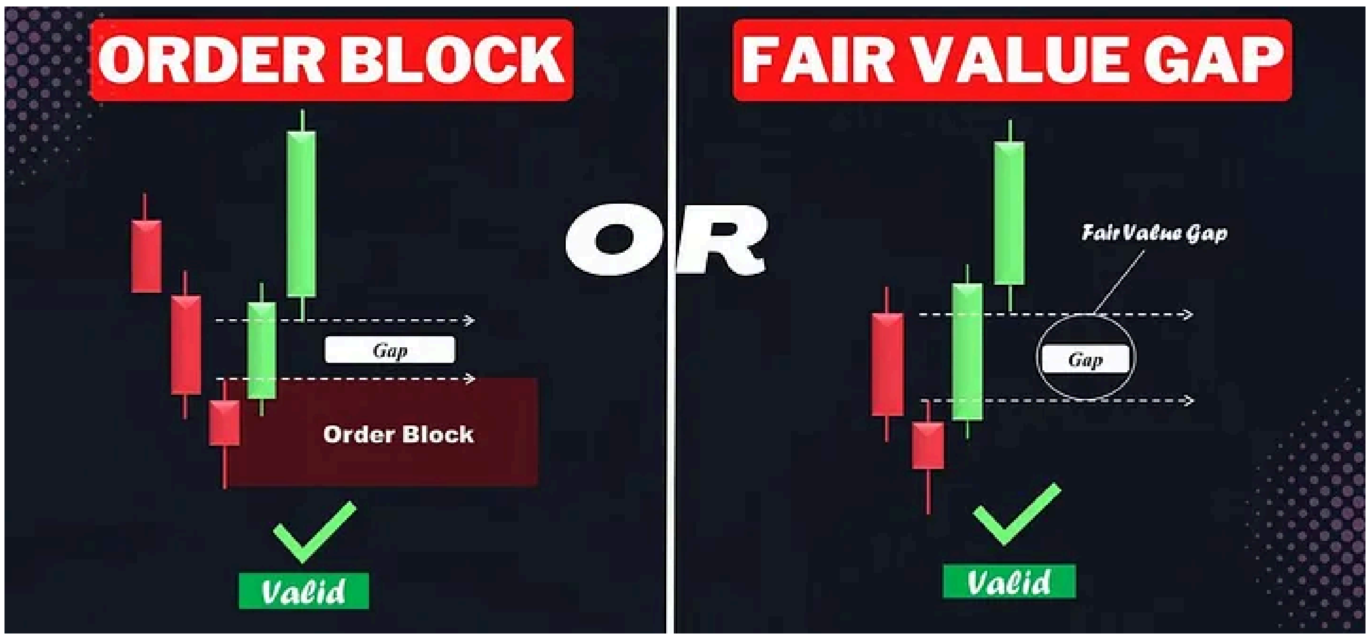
3. Waiting for Inducement (H4)



Inducement Area

This is a critical confirmation step. Before looking for an entry, the bot must see the price take out a recent, minor swing point. This action, known as taking Inducement, suggests that liquidity has been engineered and the larger move is ready to begin. It's a powerful filter against premature entries.

4. Pinpointing the Entry (M30/H1)



Order Blocks and Fair Value Gaps

Only when the first three high-level conditions are met does the bot zoom into the M30 and H1 charts. Here, it scans for high-probability entry zones: **Fair Value Gaps (FVGs)** or **Order Blocks (OBs)** that align with the established directional bias.

Pillar 3: The AI Edge — A Machine Learning Gatekeeper

An SMC setup that meets all four criteria is good, but not every setup is created equal. This is where Artificial Intelligence provides the final, critical edge.

I trained a **LightGBM (LGBM) Classifier model** on years of historical data, teaching it to distinguish the subtle characteristics of high-probability setups from those that look good but are likely to fail.

When the bot identifies a valid FVG or OB, it doesn't trade immediately. Instead, it feeds dozens of data points about the current market state into the trained model. The model acts as a final gatekeeper, returning a probability score (e.g., "I am 68% confident this setup will succeed").

> **Only if this confidence score exceeds a 55% threshold is a pending order finally placed.** This AI filter is the key to elevating the strategy from good to great, systematically filtering out mediocre setups.

Pillar 4: The Gauntlet — Testing That Mirrors Reality

A backtest can be easily manipulated to show incredible results. To ensure the bot was genuinely robust, I subjected it to a rigorous testing methodology designed to be as realistic as possible.

Walk-Forward Analysis, Not Simple Backtesting

A simple backtest on 10 years of data is prone to "overfitting." The bot might perfectly learn the past, but fail in the future. I used **Walk-Forward Analysis** to prevent this:

1. **Train:** The model trains on a 5-year window of data.
2. **Test:** It trades on the next 6 months of "unseen" data.
3. **Slide & Repeat:** The 6-month test window is added to the training data, and the entire process slides forward.

This iterative cycle ensures the bot is constantly adapting and being validated against new market conditions, providing a much more honest assessment of its performance.

Simulating Prop Firm Rules

The backtest was designed to pass a prop firm challenge, enforcing strict risk management:

- **Fixed Initial Capital:** $6,000.
- **Strict Drawdown Limits:** A hard 8% total drawdown limit. If breached, the test fails.
- **Disciplined Risk:** Each trade risked a fixed 0.5% of the *initial* capital, preventing unrealistic hyper-compounding.
- **Realistic Payout Cycle:** The simulation didn't just accumulate profit endlessly. It mirrored a prop firm's bi-weekly payout system. Every two weeks, any profit generated was "withdrawn," and the trading equity was

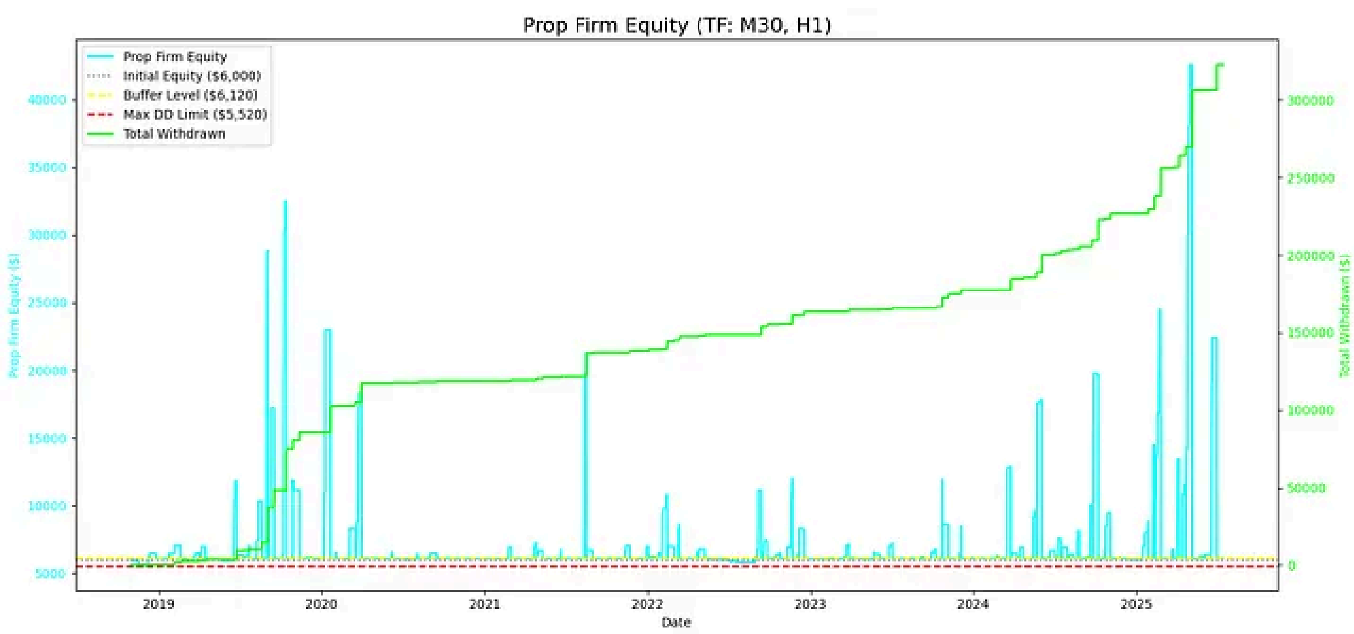
reset back to the initial capital of $6,000. This is crucial because it tests the strategy's ability to consistently generate profit from a fixed capital base, preventing the unrealistic snowball effect of hyper-compounding.

The Final Result: A Data-Driven, Disciplined Strategy

This rigorous, multi-layered filtering process is what led to the final, impressive backtest results. It's a testament to a system built on logic, validated by data, and stress-tested against realistic constraints.

When reading the report below, keep the prop firm simulation in mind: the **"Total Withdrawn"** figure represents the actual profit generated over the years, while the **"Final Capital"** remains close to the initial amount precisely because of this bi-weekly payout and reset system.

Final Backtest Report



Equity Graph over Test Period

- **Test Period:** 2018–10–29 to 2025–07–15
- **Initial Capital:** $6,000.00
- **Total Net Profit (Gross):** $322,885.95 (**5381.43%**)
- **Total Trades:** 740 (M30 + H1)
- **Win Rate: 70.32%**
- **Profit Factor: 65.06** (This means for every $1 of risk, the system generated $65.06 in gross profit)
- **Maximal Drawdown:** 7.00% of initial capital (Well within the 8% limit)

The Secret Sauce Isn't the Code, It's the Process

The journey of building this bot underscores a key principle: successful algorithmic trading is not about finding a secret indicator or a few magic lines of code.

It's about building a logical, disciplined, and multi-layered system where each component acts as a filter, ensuring that only the absolute highest-probability setups are taken. It's a process of combining a sound trading methodology with data-driven optimization and, most importantly, validating it all against the toughest, most realistic market constraints you can design.

Resources, References, and Further Reading

Core Concepts & Strategy

1. **Historical Data:** The data used for backtesting was sourced from Forexsb.
2. **Smart Money Concepts (SMC):** Investopedia's explanation of "Smart Money" provides a good starting point for the underlying theory.
3. **Market Microstructure (Academic):** Madhavan, A. (2000). *Market Microstructure: A Survey*. A foundational paper on how market mechanisms influence price formation. (Link to paper)
4. **Price Action & Order Flow:** Wyckoff, R. D. (1937). *The Richard D. Wyckoff Method of Trading and Investing in Stocks*. A classic text on analyzing market phases and institutional intent.
5. **Liquidity & Inducement:** For a deeper dive into modern liquidity concepts, resources from The Inner Circle Trader (ICT) are widely studied, though not academic.

Machine Learning & AI

1. **LightGBM Framework:** The official documentation for the LightGBM library, the high-performance tool used for the AI filter.
2. **LightGBM (Academic):** Ke, G., et al. (2017). *LightGBM: A Highly Efficient Gradient Boosting Decision Tree*. The original research paper introducing the algorithm. (Link to paper)
3. **Machine Learning in Finance (Academic):** Gu, S., Kelly, B., & Xiu, D. (2020). *Empirical Asset Pricing via Machine Learning*. A comprehensive review of how ML techniques are applied to financial forecasting. (Link to paper)
4. **Feature Engineering:** De Prado, M. L. (2018). *Advances in Financial Machine Learning*. Chapter 8 provides excellent insights into creating meaningful features from financial data.

5. **Python for Algorithmic Trading:** Chan, E. (2017). *Machine Trading: Deploying Computer Algorithms to Conquer the Markets*. A practical guide on the implementation of trading systems.

Validation & Risk Management

1. **Walk-Forward Analysis:** An Introduction to Walk-Forward Optimization from Investopedia.
2. **The Dangers of Backtesting (Academic):** Bailey, D. H., Borwein, J. M., Lopez de Prado, M., & Zhu, Q. J. (2014). *The Probability of Backtest Overfitting*. A crucial paper on why most backtested strategies fail. (Link to paper)
3. **Trading System Evaluation:** Pardo, R. (2008). *The Evaluation and Optimization of Trading Strategies*. A definitive guide on robustly testing and validating trading systems.
4. **Risk Management:** Vince, R. (2009). *The Handbook of Portfolio Mathematics*. A core text on position sizing and risk management models beyond simple fixed percentages.
5. **Overfitting and Data Snooping (Academic):** Harvey, C. R., & Liu, Y. (2015). *Backtesting*. A paper discussing the statistical pitfalls and biases inherent in backtesting financial strategies. (Link to paper)

Forex Trading | Algorithmic Trading | AI | Quantitative Finance | Commodities

A 70% win rate and 5,381% return in a backtest is the trading equivalent of telling people you bench-pressed a truck, it makes for a great story, but in the real world physics (and markets) don't work that way. Dressing it up with Smart Money... more

3 Reply

---

ajmalshamsudeen
Aug 9

this would be a wonderful work and how can i contact you and get to know about this . is this open source or is there any way to get to now you ig youtube anything...

Reply

Ashmpatel
Aug 1

Very interesting. Is it possible to get the code ?

Reply

Recommended from Medium

Sayedali

This 20-Period Trick Shows Exactly When to Buy the Dip

Most traders spend months trying to catch trend reversals. But smart traders know—th...

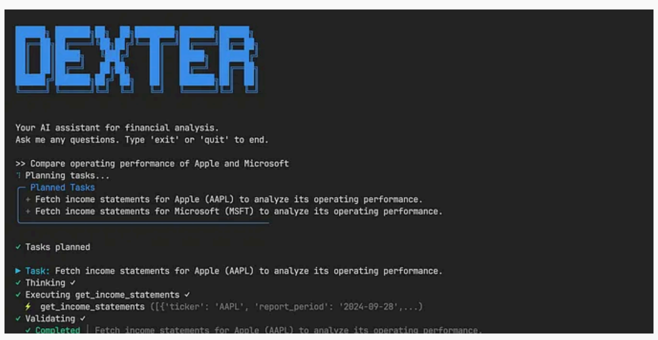
Nov 11 57

In InsiderFinance Wire by Brian Hulela

6 Ways AI Trading Algorithms Are Rigged Against You

How institutional algorithms exploit retail traders and what you need to know to survive

4d ago 132



In Generative AI by Adham Khaled

In Inside The Trade by RB Trading

I Built a Wall Street Analyst in 200 Lines of Code — And It...

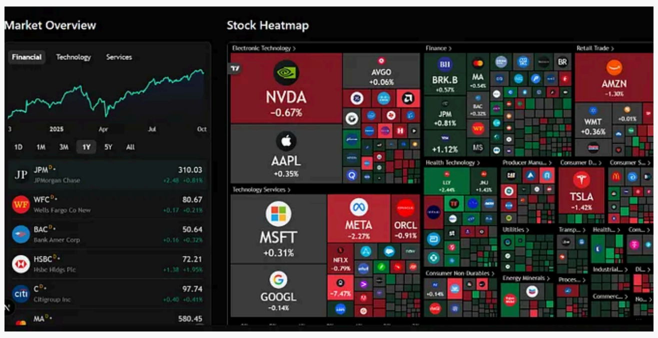
How an open-source AI agent named Dexter is democratizing financial research, one...

Oct 20 · 381 · 10

The Two Moving Averages That Turned $100K Into $1M

Why Stock Millionaires Only Buy Pullbacks

Oct 22 · 318 · 14

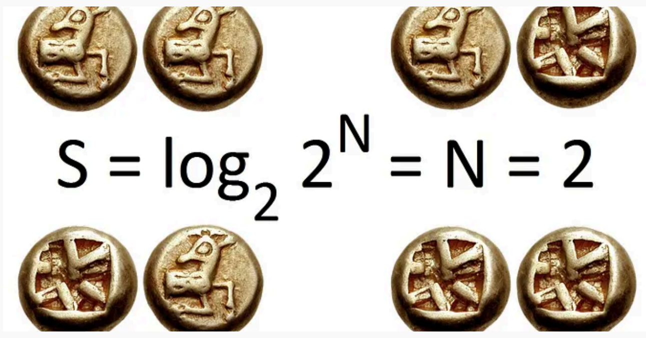


In Coding Nexus by Tattva Tarang

OpenStock — The Free, Open-Source Stock Market App Built fo...

Some time ago, I noticed I was spending more time switching between finance apps than...

Oct 19 · 60



In CodeX by Richard Shu

Trading with Less Surprise: Using Shannon Entropy to Improve a...

How a concept from information theory, implemented in the AlphaSuite open-source...

3d ago · 24

See more recommendations