

the WiseBot

Failed

WISEBOT • VERSION 1

WisePolyBot

A bot that won 90% of its bets and still lost money.

First in a series documenting every version of WiseBot — what each one did, and why it failed or worked. This one failed. Here is the full autopsy, numbers included.

the WiseBot

the-wisebot.com
June 2026

Who I am, and why I'm writing this

Let me say who I am first, because it changes how you should read the rest. I'm anonymous, and I'm not a finance professional — no trading desk, no quant background, no credentials to wave at you. I'm someone who likes investing and likes pulling technology apart to see how it works. That's the whole résumé. What I *do* have is the habit of writing down exactly what happened, the parts that went wrong included. That habit is the entire point of this project.

WiseBot is not a signal service and not a get-rich pitch. It's a public, version-by-version record of one person trying to build a trading system that actually holds up — and being honest, in writing, about every version that didn't. This is version one. It didn't.

I'm telling you this story for a simple reason: anyone can show you their winning trades. The thing that's rare, and the only thing I can offer that's worth your trust, is a complete and verifiable account of the losing ones too.

The hunch

WiseBot started as a hunch. I'd been poking around Polymarket's BTC 5-minute "Up/Down" markets — binary bets that resolve all-or-nothing every five minutes — and I noticed something: in the final seconds of a window, once one side drifted up to around \$0.90, it almost always resolved that way. It looked like free money.

So I did the obvious thing for someone who likes both investing and code — I tried to automate the hunch. That bot was WisePolyBot — *WiseBot* for short, the name I'll use from here on.

The build

I'm going to keep the mechanics plain, because the lesson here isn't technical.

The bot watched each 5-minute market and did nothing until the last seconds of the window. Then it bought the favored side — the one the crowd had already pushed to around \$0.94–0.96 — and let the market resolve. A binary market settles to either \$1.00 (you were right) or \$0.00 (you were wrong). Buying at \$0.96 means paying 96 cents for something that pays back one dollar if it wins.

It ran live, on a small server, on its own wallet, for about a month, after a paper-trading phase to shake out the plumbing. It went through a few iterations of how exactly it placed and chased orders in those final seconds. None of that mattered to the outcome, which is itself part of the lesson.

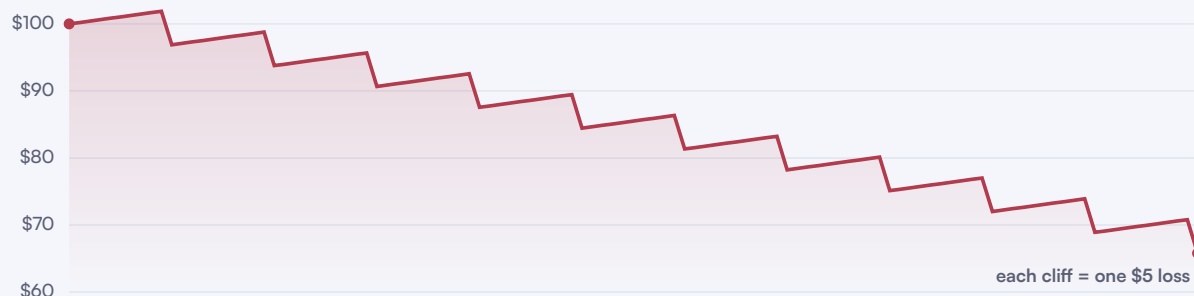
The results

The bot won roughly **90% of the time**.

If you stop reading there, it sounds like a success. Nine bets out of ten resolved in my favor. The equity curve still went down.

The bleed: 90% wins, still relentlessly downhill

Illustrative — the documented mechanism, not logged ticks.



Illustrative. v1 ran on Polymarket and its per-trade equity isn't preserved in a verifiable log here, so this is the documented shape, not logged data: nine ~\$0.21 wins, then one ~\$5 loss that erases them — repeated. Ninety percent of the steps go up; the curve still goes down. The slope is the whole point.

The autopsy: why it failed

It lost to arithmetic I hadn't respected.

At a \$0.96 entry, a \$5 winning bet returns about **+\$0.21**. A losing bet returns nothing: **-\$5.00**. That is a reward-to-risk ratio of roughly **1:24 against**. To break even at those odds you need to win about **96%** of the time. WiseBot won 90%. Each loss erased roughly two dozen wins.

Why a 90% win rate still lost money

Buy the favorite at \$0.96 — it pays \$1.00 if right, \$0.00 if wrong.

On a single \$5 bet

Win **+\$0.21**

Loss **-\$5.00**

Reward-to-risk $\approx 1 : 24$ — one loss wipes out about two dozen wins.

Win rate

Achieved **90%**

Break-even **96%**

The 6-point gap between 90% and 96% is exactly where the money went.

The market price **is** the odds: \$0.96 prices a ~4% chance of loss, and on a reasonably efficient market that 4% shows up about as often as priced. A high hit rate was never an edge.

This was not a bug. The code did exactly what I told it to. It was a *design* error — the kind no amount of debugging fixes, because the expected value was negative before the first line of code ran.

Here is the part worth carrying out of this document: **a high win rate is not an edge**. The market offering \$0.96 *is* the market pricing a real ~4% chance of loss — and on a reasonably efficient market, that 4% shows up exactly as often as priced. I had mistaken a high hit rate for a profitable system. They are not the same thing, and the gap between them is precisely where the money went.

What an edge would have looked like

For this strategy to actually work, I would have needed a **mispricing**: situations where the true probability of loss was lower than the ~4% the market was charging me for. That's a real, findable thing — but it has to be measured and proven, not assumed.

WiseBot never looked for that. Nothing in its design tested whether \$0.96 prices were *systematically* too high. It simply bet that they were, on a gut feeling, and a gut feeling is not an edge. The honest verdict is that I built an efficient way to execute a bet I had never validated.

What I took away

Three things came out of this that every later version inherits.

1. **Win rate alone means nothing.** Before any metric of "success," the first question is: given what the market is charging, is the expected payoff positive? If the answer is no, a 99% win rate still loses.
2. **A structural failure is not a technical one.** When a system loses money, the first suspect is the strategy's math, not the code. I spent effort polishing execution that was irrelevant to a doomed payoff structure.
3. **No real capital before a validated edge.** A good-looking live result on a short run can be pure luck of a single market regime. Proof has to come first — and it has to come from something more honest than "it's been working lately."

WiseBot v1 was switched off. The next version moved to a different market with a healthier payoff geometry, carrying these three rules forward. It also failed — for new reasons, which is the next paper.

Methodology note

The analysis and drafting of this document are AI-assisted (I work with Claude as my stated method, the same way the later bots use AI). Every figure here is drawn from the bot's own logged trade data; every claim is mine and is meant to be verifiable. The Polymarket account that placed these trades is public — see polymarket.com/@wisebot. The AI helps me write clearly and check my reasoning — it does not invent the numbers, and it does not get to decide what's true. That responsibility stays with me.

the WiseBot

Verify me, don't trust me.

the-wisebot.com

Not financial advice. Past performance does not indicate future results.