1. Introduction

I started Artemis to go beyond “I coded something with ChatGPT” and build a trading strategy that actually worked — or at least taught me something real about market behavior. I chose crude oil because it’s a high-volatility asset with global importance and unique supply/demand dynamics, unlike more common assets like equities or ETFs. My goal was to test whether basic technical indicators could still perform reasonably well on a commodity asset, and to learn the full stack of data analysis, strategy logic, and performance evaluation.

---

2. Strategy Design

The core of Artemis is built around two indicators:

- **Relative Strength Index (RSI):** Used to identify overbought/oversold conditions.
- **Moving Averages:** A 10-day simple moving average (SMA) and 30-day SMA crossover.

Buy Signal:

- RSI < 30 (oversold)
- 10-day SMA crosses **above** 30-day SMA

Sell Signal:

- RSI > 70 (overbought)
- 10-day SMA crosses **below** 30-day SMA

The strategy only enters a position when both RSI and crossover conditions are met. This reduces false positives and adds signal confirmation.

---

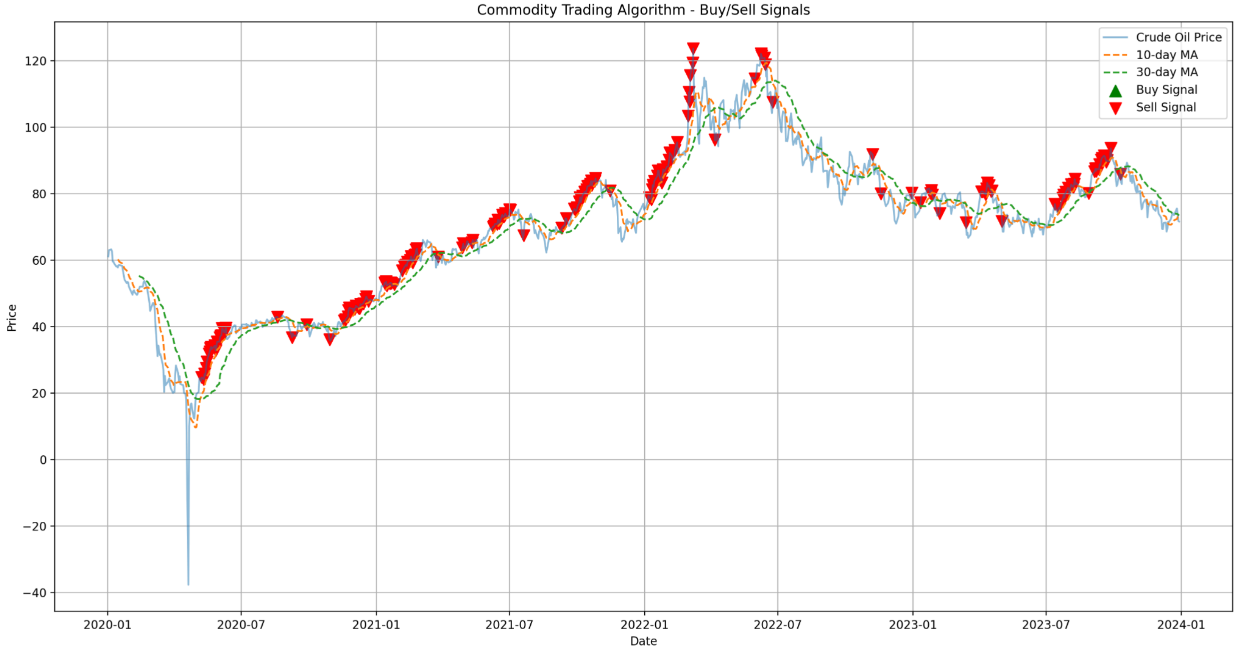
3. Backtesting Setup

I used the `yfinance` Python library to pull daily crude oil price data from 2020 to 2024. All data processing and signal generation was handled using pandas and numpy. Trades were tracked in a custom simulation loop — no third-party backtesting libraries — to ensure I understood how every trade was triggered and closed.

Visualization was done in `matplotlib`, where I plotted buy/sell points directly on the price chart to verify logic and timing.

4. Results

Across selected 3-month test windows, Artemis achieved up to **~20% simulated ROI**, though performance varied depending on market conditions. In flat or choppy markets, signals led to smaller gains or break-even trades. The strategy worked best in trending environments with clear momentum reversals.



The chart above shows one of the test windows with visualized entry and exit points. Trades aligned with trend reversals in most cases, validating the strategy's logic.

5. Future Work

Next, I'm deploying Artemis into **live paper trading** using the Alpaca API, so I can simulate real-time execution with fake capital but real price data. I also plan to expand the strategy to support:

- **ETFs** like SPY and QQQ

- **Additional indicators** like Bollinger Bands or volatility filters
- A simple **machine learning classifier** (e.g. Random Forest) to predict next-day signals based on current metrics

---

6. Conclusion

Building Artemis helped me see that algorithmic trading isn’t magic — it’s just disciplined logic applied to real data. Writing and testing every line of this bot taught me more about finance, Python, and statistical thinking than any class I’ve taken. More importantly, it gave me a framework for improving, iterating, and sharing my work — not just coding for myself, but building something that can be tested, explained, and eventually expanded.
