# AI Trading Strategist

IMPORTANT: This is a paper trading project using Alpaca's simulated trading API. No real money is involved. Do not connect real brokerage accounts or use real funds. This is an educational exercise in building AI systems, not a path to trading profits.

## Why This Matters

Financial markets are one of the most challenging domains for AI because the system fights back. Unlike chess or protein folding where the rules are fixed, markets are full of intelligent agents who adapt to any exploitable pattern. A strategy that works today may fail tomorrow precisely because it worked today. Economists call this the Adaptive Market Hypothesis.

Building a trading system forces you to grapple with uncertainty, feedback loops, and the difference between correlation and causation. Every mistake shows up in the P&L. There is no hiding behind vague success metrics when your Sharpe ratio is negative.

This project is explicitly educational. Paper trading only. Anyone who claims they can build a consistently profitable system in a semester is either lying or about to learn an expensive lesson. The value is in the process: building systems that reason about uncertain, adversarial environments.

## The Product Vision

The AI Trading Strategist connects to market data via Yahoo Finance and similar APIs, analyzes historical patterns, generates trading hypotheses, and backtests strategies across configurable time horizons (12, 24, 36, 72, 144 months). All trades execute through Alpaca's paper trading API, so no real money is at risk.

What distinguishes this from a simple backtesting framework is the AI's role in hypothesis generation. Instead of the human specifying "test a momentum strategy with these parameters," the AI proposes strategies based on its analysis. The human shifts from strategy designer to strategy evaluator and risk manager.

The system compares performance against standard benchmarks: S&P 500, NASDAQ 100, 60/40 portfolio, and risk-free rate. Risk-adjusted metrics (Sharpe ratio, max drawdown, win rate) matter more than raw returns.

## Agentic Architecture

Data Aggregation Agent collects, cleans, and normalizes market data from multiple sources. This is harder than it sounds: different providers handle stock splits, dividends, and corporate actions differently. Yahoo Finance has known data quality issues that can corrupt backtests if not caught.

Strategy Generation Agent analyzes processed data to identify trading opportunities. This is where LLMs get interesting. Traditional quant approaches use predefined factors (momentum, value, mean reversion). An LLM can potentially identify subtler patterns by reasoning in natural language, like noticing a sector tends to outperform after certain Fed announcements.

Backtesting Agent evaluates strategies against historical data with rigorous safeguards against lookahead bias (using information not available at the time) and survivorship bias (only including companies that still exist). Proper backtesting is an art that students consistently underestimate.

Risk Management Agent enforces constraints on position sizes, sector concentrations, and drawdown limits. This agent has veto power: no matter how attractive an opportunity looks, if it violates risk limits, it does not execute. This separation mirrors best practices at real trading firms.

Paper Trading Agent interfaces with Alpaca to execute simulated trades, handling order types, partial fills, and API failures gracefully.

## Implementation Hints

Regime change is the killer. A momentum strategy that worked in 2017-2019 might have been catastrophic in March 2020. Build an ensemble of strategies with different characteristics and dynamically adjust allocations based on what is currently working.

Transaction costs matter. A strategy that looks profitable before costs may be unprofitable after accounting for commissions, bid-ask spreads, and market impact. Model these realistically.

Use walk-forward analysis. Train on months 1-12, test on month 13. Retrain on months 1-13, test on month 14. This simulates real-time performance including periodic retraining effects.

Out-of-sample testing is mandatory. If you tune parameters on the same data you test on, you are fooling yourself. The strategy will look great historically and fail immediately in production.

Start simple. A basic momentum or mean-reversion strategy that you understand deeply beats a complex ML model that you cannot explain. Add complexity only when you have exhausted simple approaches.

## Interesting Questions

How does LLM-based hypothesis generation compare to traditional factor models? Are there patterns that language models catch that quant screens miss, or vice versa?

What is the right response to drawdown? Fixed rules (stop after 10% loss) are common but crude. Can an AI distinguish between bad luck within an intact strategy versus evidence that the strategy is broken?

How do you handle explore-exploit? Deploy your best strategy (exploit) or keep experimenting (explore)? This is a multi-armed bandit problem where the arms' payoffs are non-stationary.

Can the AI learn from its mistakes? When a strategy fails, can it perform post-mortem analysis that prevents similar failures?

## Market Data Sources

****Yahoo Finance**** ([yfinance](https://github.com/ranaroussi/yfinance)) provides free historical data going back decades for most US equities and indices. S&P 500 data is available from 1927, individual stocks typically from their IPO date. Daily OHLCV (open, high, low, close, volume) data is reliable; intraday data is more limited. Be aware of survivorship bias: Yahoo only has data for companies that still exist or were acquired, not those that went bankrupt.

****Alpha Vantage**** offers a free tier with historical and real-time data. More reliable than Yahoo Finance for some use cases, but rate-limited on the free plan. Good for validating Yahoo Finance data quality.

****FRED (Federal Reserve Economic Data)**** provides macroeconomic indicators: interest rates, unemployment, GDP, inflation. Useful for regime detection and macro-aware strategies. Available via the [fredapi](https://github.com/mortada/fredapi) Python library.

****Alpaca Markets**** provides both historical data and paper trading execution through a single API. Their free tier includes 5+ years of daily data and real-time quotes during market hours. This simplifies the architecture since you can get data and execute trades through one integration.

****Kaggle Datasets**** host several curated financial datasets for research, including historical S&P 500 constituents (addresses survivorship bias), earnings announcements, and sentiment data. Useful for specific research questions.

For backtesting, 10-20 years of data is typically sufficient to capture multiple market regimes (bull markets, bear markets, high volatility periods like 2008 and 2020). Going back further (50+ years) introduces questions about whether historical patterns remain relevant given structural changes in markets.

## Boundaries

Paper trading only. Mandatory, not optional. Real money creates inappropriate pressures for a learning environment.

This is not financial advice. Past performance does not guarantee future results. Even sophisticated AI systems at well-funded firms frequently fail. Renaissance Technologies is the exception; most quant funds have mediocre records. Humility is appropriate.