Replicating corporate insider alpha via Machine Learning

Building an autonomous trading agent to outperform the S&P 500

Charles Husson

February 2026

1 Overview

1.1 From Congress to corporate insiders

Recently, copy-trading US Congress members has become highly popularized, leading to dedicated tracking websites and even public ETFs. However, digging deeper into insider data reveals a much larger and more actionable opportunity with corporate insiders (CEOs, CFOs, and board members). For algorithmic trading, congressional data suffers from a fatal technical flaw: a massive reporting delay of up to 45 days. In contrast, corporate insiders are legally bound by the SEC to report their trades within two business days via Form 4. To capture price movements before they are fully absorbed by the broader market, the scope of this project was shifted exclusively to these low-latency corporate disclosures.

1.2 Problem statement

The stock market is fundamentally an environment of information asymmetry. Retail traders operate on delayed, public news, while corporate insiders possess material, non-public information about their own companies' health and future earnings. While tracking these insiders offers a theoretical edge (often called "alpha," representing excess returns relative to a benchmark index), isolating the true high-conviction trades from the noise is difficult.

- **Data ingestion & fidelity:** Processing massive volumes of raw, unstructured SEC filings while filtering out institutional "drip-feed" noise (where single large orders are executed as hundreds of micro-trades).
- **The 2-day lag:** The SEC requires Form 4 to be filed within two business days. If the market prices in the insider's action during that window, the trade's alpha decays rapidly.
- **Non-linear patterns:** A trade's significance depends on a complex web of factors (e.g., the executive's role, historical routine, and the company's market cap) that simple linear formulas cannot accurately capture.

1.3 Project goal

The goal is to build a program that can autonomously select and weight trades from corporate insider data, and copy-trade them on a real portfolio. To succeed, the system must consistently outperform the S&P 500 over a multi-year period. This is achieved by training a machine learning model on 15 years of public insider trading databases to identify the precise conditions that lead to excess returns.

2 Technical architecture

The solution is designed for offline efficiency, processing 15 years of market data locally to train the predictive model before execution.

2.1 System pipeline and codebase

The project's architecture is strictly modular, separated into distinct execution phases to prevent data leakage. The local environment (`Historical_Research/`) contains the following core scripts:

- **Data extraction:** `insider_scraper.py`, `market_data_fetcher.py`, `generate_full_db_baseline.py`, and `processor_cleaning.py` handle SEC EDGAR and Yahoo Finance API connections, liquidity filtering, and database construction.
- **Feature engineering:** `feature_engineering.py` merges the datasets, aggregates micro-trades, and calculates the quantitative ML features to compile the master matrix.
- **ML & Backtesting:** `train_xgboost.py` trains the model, while `explain_model.py` opens the "black box" via SHAP visualizations. Finally, `portfolio_backtest.py` and `plot_performance.py` run the capital-constrained Kelly simulations. (`trading_agent_groq.py` is reserved for LLM reasoning interfaces).

2.2 Data warehouse layer

- **SEC EDGAR database:** Bulk extraction of raw Form 4 disclosures from 2010 to 2025. Crucially, "Sell" signals (Code 'S') are entirely excluded from the dataset, as insiders sell for various personal liquidity reasons but buy for only one: expected capital appreciation.
- **Micro-trade aggregation:** To prevent artificial inflation of trading signals, intraday order-splitting (where a broker fills a block order via hundreds of smaller transactions) is mathematically aggregated into a single daily consensus row per executive.
- **Tradability filters:** To ensure the AI only trains on realistic setups, strict liquidity thresholds were applied (minimum $2.00 share price and ¿50,000 average daily volume), effectively banning untradable penny stocks and micro-cap noise.

2.3 Feature engineering

The pipeline explicitly calculates 8 strictly numerical metrics (`FEATURES`) to feed the machine learning algorithm:

- **Close & Volume:** Captures the asset's market-cap tier and raw baseline liquidity.
- **ATR_14:** 14-day Average True Range, measuring the stock's recent volatility and swing capacity.
- **Value:** The absolute dollar amount committed by the insider in the transaction.
- **Pct_Volume_Absorbed:** The trade's dollar value divided by the stock's average daily dollar volume.
- **Portfolio_Pct:** Skin in the game (shares bought divided by total shares owned post-transaction).
- **Role_Weight:** Hierarchical scoring of the executive's title (e.g., CFOs receive higher weights than standard Directors).
- **Consensus_Score:** A rolling weighted score identifying simultaneous "wolfpack" buying across the board within a 7-day window.

3 Algorithmic strategy & machine learning

Rather than manually defining a rigid formula with arbitrary weights, this project utilizes supervised machine learning to autonomously discover the optimal trading logic, outputting probabilistic confidence levels rather than simple binary signals.

3.1 Probabilistic target labeling

While the algorithm is trained on historical binary profitability (where 1 equals a positive return over the target holding period and 0 equals a loss), the XGBoost classifier is configured to output a continuous probability score. This score represents the model's statistical confidence that a specific insider trade will generate alpha, allowing the downstream engine to exclusively execute high-conviction setups.

3.2 XGBoost and SHAP analysis

An XGBoost (extreme gradient boosting) classifier is trained on the engineered feature matrix. Post-training, SHAP (SHapley Additive exPlanations) is used to interpret the model. This game-theory approach reveals which underlying features drive the highest confidence scores, ensuring the model's logic is financially sound. As shown in Figure 1, the model relies heavily on market microstructure (`Close`, `Volume`) to capture short-term momentum, while utilizing `ATR_14` and insider fundamentals (`Consensus_Score`) as secondary filtering mechanisms.

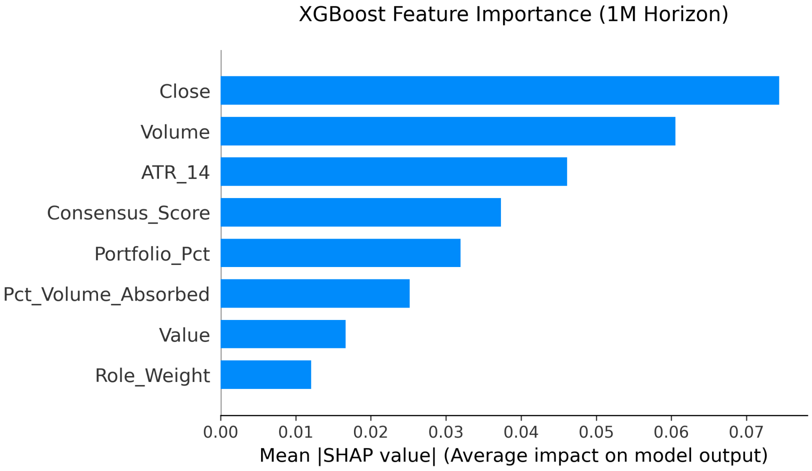

Figure 1: Mean SHAP feature importance (1-Month Horizon) showing the model's reliance on short-term market microstructure (Close, Volume) to capture immediate momentum.

3.3 Dynamic portfolio allocation and risk constraints

Raw machine learning accuracy does not directly translate to financial returns. To evaluate the true Compound Annual Growth Rate (CAGR), the model's probabilities are passed through a rigorous, capital-constrained financial backtester that mirrors real-world trading conditions.

- **Capital-constrained execution:** The simulation enforces a strict daily cash budget, preventing the "infinite leverage" fallacy common in basic data science backtests. If the algorithm generates multiple high-confidence signals but the portfolio lacks available liquidity, the trades are rejected.

- **Kelly Criterion Optimization:** To maximize geometric portfolio growth while managing drawdown, dynamic position sizing was calculated using the Half-Kelly fraction. To mitigate single-asset concentration risk, an absolute maximum risk ceiling of 5% per trade was enforced.
- **Horizon Selection:** The backtester evaluated 1-Month, 2-Month, and 6-Month holding periods (Table 1). While the 6-Month model offered extreme capital preservation (-2.8% Drawdown), the 1-Month momentum model was selected as the optimal deployed strategy, identifying an optimal confidence threshold of 60% to drive an aggressive 144.7% True CAGR.

Horizon	Threshold	True CAGR	Max DD	Sharpe	Win Rate	Trades
1-Month (21 Days)	60%	**144.7%**	-16.7%	5.51	59.1%	3,239
2-Month (42 Days)	34%	93.1%	-10.1%	3.39	55.1%	1,755
6-Month (126 Days)	60%	58.3%	**-2.8%**	2.42	60.6%	548

Table 1: Optimal performance metrics across time horizons. The 1-Month model maximizes pure growth (CAGR), while the 6-Month model prioritizes extreme capital preservation (Drawdown).

4 Code Availability & Live Deployment

The historical data extraction, feature engineering pipelines, and backtesting engine are open-source and available on GitHub at :
`https://github.com/charleshusson75-cell/corporate-insider-momentum-research`.
To protect proprietary Alpha and hyperparameter configurations, the live event-driven execution architecture and the serialized XGBoost model are maintained in a secure, private repository. The live system autonomously parses SEC XML feeds and executes position-sized trades via the Alpaca API. For inquiries regarding the live architecture or commercial data access, please contact me.

5 References

- **Securities and Exchange Commission (SEC).**
  "Form 4: Statement of changes in beneficial ownership".
  The legal backbone of this dataset. Section 16(a) of the Securities Exchange Act of 1934 mandates that insiders must report open market purchases within two business days, providing the strict, low-latency framework required for this strategy.

- **Cohen, L., Malloy, C., & Pomorski, L. (2012).**
  "Decoding Inside Information". The Journal of Finance, 67(3), 1009-1043.
  The foundational paper establishing the "routine vs. opportunistic" framework. It demonstrates that filtering for irregular trading patterns generates significant alpha, whereas routine trades contain no predictive signal.

- **Wang, W., Shin, Y. C., & Francis, B. B. (2012).**
  "Are CFOs' trades more informative than CEOs' trades?"
  Journal of Financial and Quantitative Analysis, 47(4), 743-762.
  Provides empirical evidence that CFOs possess deeper insights into short-term earnings trajectories than CEOs, directly influencing the hierarchical feature weighting in the machine learning model.

- **Ahern, K. R. (2017).**
  "Information networks: Evidence from illegal insider trading tips".

Journal of Financial Economics, 125(1), 26-47.
Analyzes how information flows through executive networks, supporting the project's consensus feature logic, where clustered buying signals stronger asymmetric information.

- **Lynch, P., & Rothchild, J. (1989).**
"One Up on Wall Street: How to use what you already know to make money in the market".
Simon & Schuster.
Coined the fundamental axiom of insider tracking: insiders sell for countless personal reasons (portfolio diversification, tax obligations, liquidity needs), making sell signals statistically noisy. Conversely, open-market purchases strictly indicate expected price appreciation, justifying the exclusion of SEC 'S' (Sell) codes from the training data.