Predicting Stock Price Movements Around Earnings for NVIDIA, Apple & Google in the Era of AI & Trade Tensions: Final report

Apollo 11: Carlson Nkwain, Giuliana Pauletti, Jonathan Wise, Lakmali Amarawardana, Liam Hammond & Tehila Sharabi

LSE THE LONDON SCHOOL OF ECONOMICS AND POLITICAL SCIENCE

Apollo.11

Contents

Introduction

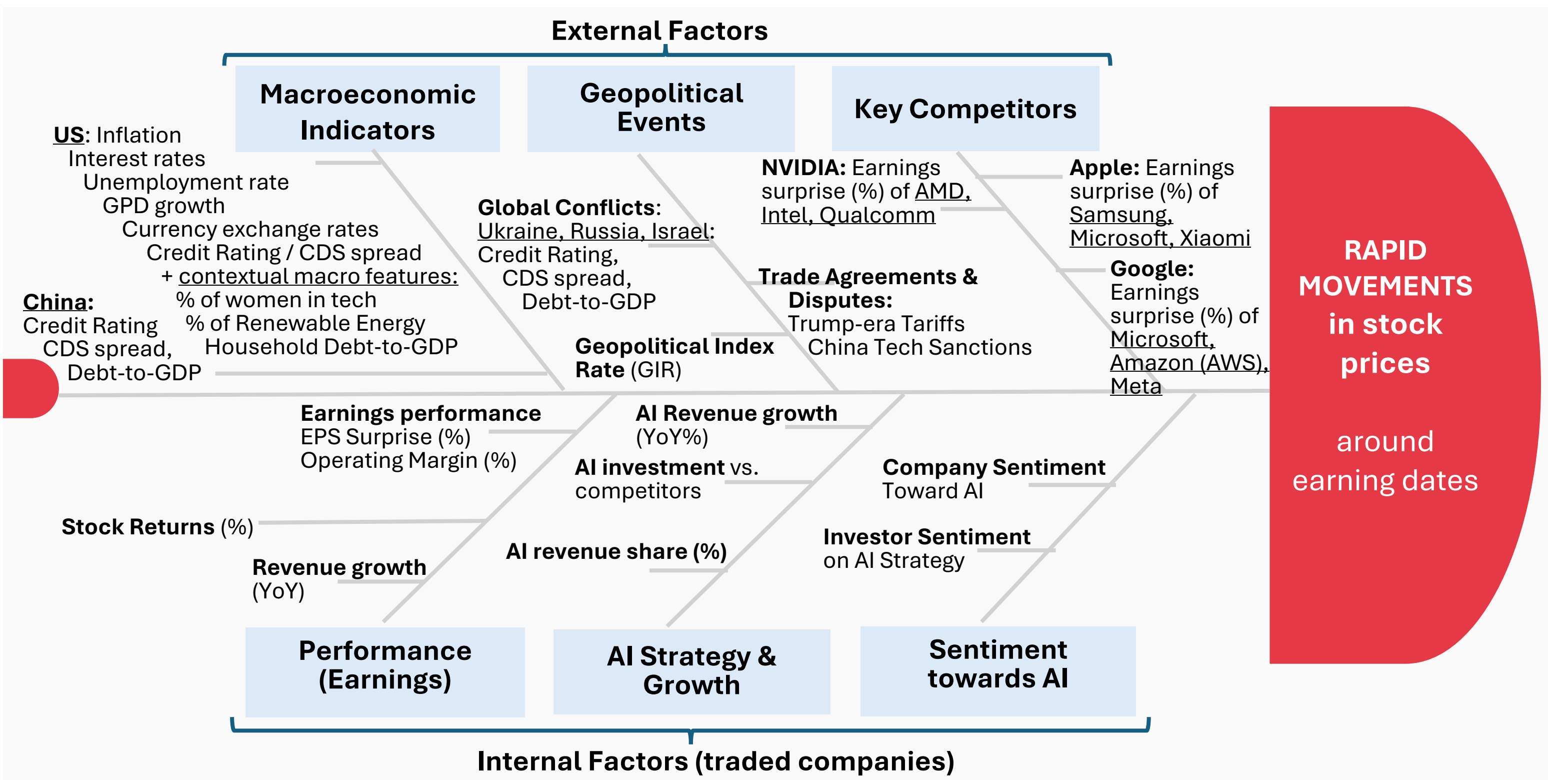
VP Analytics and the challenge of predicting rapid movements in stock prices

VantagePoint Analytics (VP Analytics) delivers Data Analytics as a Service (DAaaS) to clients across industries – with a particular focus on hedge funds – supporting data-driven, high-precision investment decisions. Competing with global leaders such as Bloomberg, S&P Global, Refinitiv, Morningstar, and FactSet, VP Analytics is seeking to strengthen its competitive edge by developing new predictive models that explain and anticipate **rapid stock price movements around earnings dates**.

Two contemporary forces were identified as especially relevant in shaping stock volatility in today's U.S.: (**a**) evolving **AI strategies and sentiment**, which signal growth potential and shape investor behaviour; and (**b**) rising **geopolitical and trade tensions**, including renewed Trump-era tariffs, which introduce heightened uncertainty and risk.

Guided by these insights, a comprehensive mapping of factors was conducted – integrating macroeconomic indicators, geopolitical risks, competitive dynamics, and company-specific performance metrics – for three key tech firms: **NVIDIA, Google, and Apple**. The resulting diagram below visualises a wide range of potential root causes behind stock price fluctuations in present-day financial markets, providing a structured foundation for model development and further testing.

Possible Causes for Rapid Movements in NVIDIA, Apple, and Google's Stock Prices



Earning Surprise (%): The percentage difference between a company's actual and expected financial results. It is most commonly calculated for earnings per share (EPS), but can also be based on revenue. Offers a clean, comparable metric across firms with diverse product lines
Operating Margin (%): The percentage of revenue left after paying for core operating costs (before interest and taxes). Formula: operating Income ÷ Revenue × 100. Preferred over profit margin as it excludes tax and financing noise.
Stock Returns (high/low): Measured over a fixed pre-event window (e.g., 7 days prior to earnings) to capture market momentum while avoiding post-event data leakage; trailing return windows of 3, 7, or 14 days can be tested.
Revenue growth (YoY): Captures internal business momentum and is widely used by hedge funds and institutional investors as a key indicator of company trajectory.
Company Sentiment Toward AI can be derived from earnings call transcripts — verbatim records of quarterly analyst calls. These transcripts reveal how companies communicate about AI, capturing tone, strategic focus, and sentiment shifts. Unlike formal filings, they reflect real-time executive positioning, making them well-suited for NLP-based sentiment analysis.

Substantial work on AI sentiment and tariff-related news via NLP was completed; however, due to limits in data quality and engineering capacity, these signals cannot currently be leveraged to improve model performance. Nonetheless, by integrating macro/geopolitical context (including trade-volume data) with firm metrics and earnings releases, and applying advanced feature engineering and modelling, **the project identifies key drivers of price movement across the year and around earnings, strengthens their predictive power, and translates them into deployable short- and long-horizon strategies.** The performance of the final predictive models supports a strong case for deployment, alongside continued refinement and optimisation of a multi-layered modelling pipeline to enable data-driven, risk-mitigated investment decisions.

Project development process

Designing a machine-learning model to predict short-term stock returns

This project aimed to develop a machine-learning model to predict short-term stock returns both around earnings announcements and year-round. The study integrated fundamental, technical, sentiment, and market signals into a unified feature set and trained an XGBoost predictor – a tree-based ensemble widely used in financial forecasting.

To ensure temporal consistency and avoid distortions from COVID-era dynamics, the analysis focused on the most recent two years of data. The dataset was constructed to examine post-earnings price reactions and volatility patterns among large-capitalization technology equities, combining multiple sources: earnings figures from the Alpha Vantage API, earnings-announcement dates from the Nasdaq API, quarterly revenue from Macrotrends, and daily price/volume for 15 selected tickers from the Yahoo Finance API. Macroeconomic data were sourced via the Federal Reserve Economic Data (FRED) API. Federal Open Market Committee meeting (FOMC interest-rate announcement) days were scraped from the Fed's website. News article headlines and summaries were scraped from Finnhub API with OpenAI API used to perform sentiment analysis on these. All data was then exported directly via Python to Postgres SQL for cleaning and transformation into a single master data file.

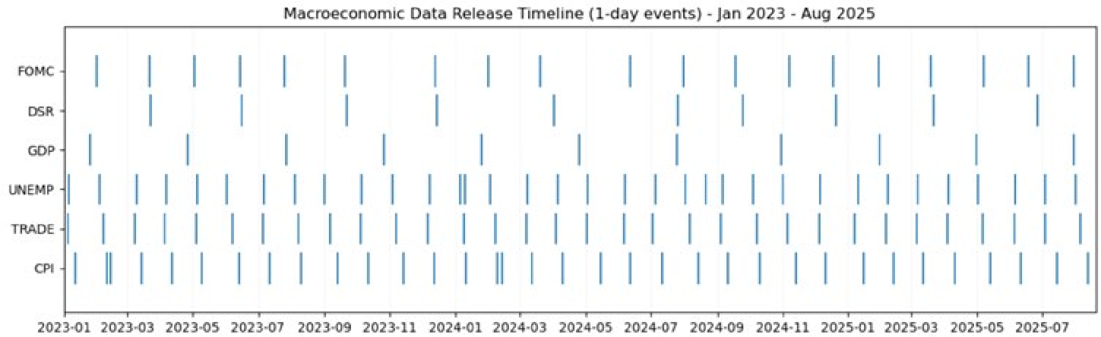
The model was designed to forecast short-horizon returns (year-round and within earnings windows), using price/volume data, technical indicators (e.g., RSI, MACD, Bollinger Bands), earnings surprises, and sentiment scores. It also incorporated macroeconomic indicators (e.g., policy rates, inflation, unemployment, debt-to-GDP) and calendar/event variables (day of week, proximity to quarter-end, earnings-window flags).

The model was trained on 3 different paths: **Path A (Peers only)** – Trained on a panel of five peer stocks selected separately for Google, NVIDIA, and Apple; **Path B (Target only)** – Trained solely on the target stock (e.g., Google); **Path C (Target + Peers)–** Trained on the combined set of the target and its peers. Model outputs are point forecasts of future returns (e.g., +3.1%) that inform trading signals—buy, hold, or sell—based on confidence levels. After evaluating results across all three paths, the optimal path was selected for each target company.

Project Development Solutions

Mixed-Frequency Alignment, Quality Assurance, and Cohort Construction

Incorporating mixed-frequency predictors in a daily panel presented two main challenges: data leakage and data flatness/staleness. To avoid leakage, all data were applied from the release day onward (forward-filled only until the next release). To mitigate flatness/staleness, event-window indicators (1-, 2-, and 5-day) and "days-since-release" columns were added to each low-frequency feature.



date	cpi_event_1d	trade_event_1d	unemp_event_1d	gdp_event_1d	dsr_event_1d	fomc_event_1d
2023-01-03 00:00:00	0	0	0	0	0	0
2023-01-04 00:00:00	0	0	0	0	0	0
2023-01-05 00:00:00	0	1	0	0	0	0
2023-01-06 00:00:00	0	0	1	0	0	0
2023-01-09 00:00:00	0	0	0	0	0	0
2023-01-10 00:00:00	0	0	0	0	0	0
2023-01-11 00:00:00	0	0	0	0	0	0
2023-01-12 00:00:00	1	0	0	0	0	0
2023-01-13 00:00:00	0	0	0	0	0	0

An illustrative example of the Event-Flag (No-Leakage) Approach, showing binary 1-day event flags on the trading-day calendar.

To ensure data quality all data were cross-referenced and validated against official sources – including Earnings Reports, Federal Reserve Economic Data (FRED), the U.S. Bureau of Economic Analysis (BEA), the U.S. Bureau of Labor Statistics (BLS), and the U.S. Census Bureau.

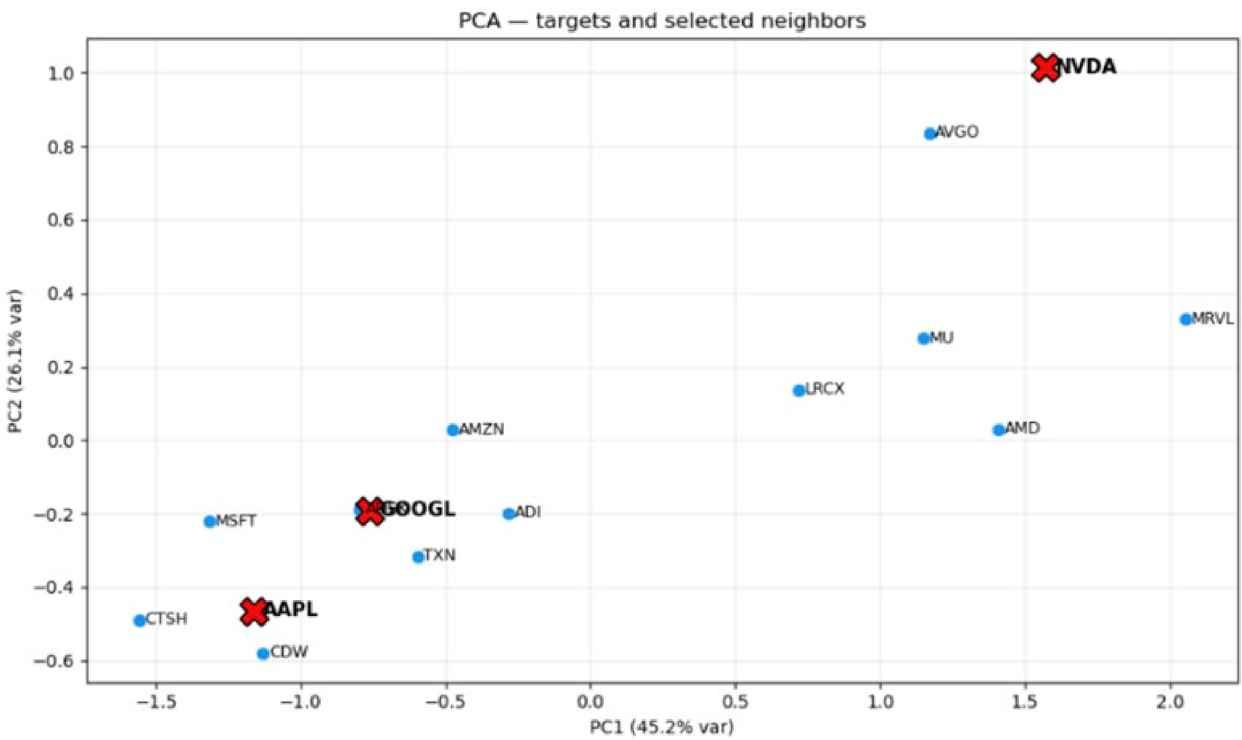
The stock-price feature set was engineered for two goals: **(i)** predictive modelling and **(ii)** unsupervised clustering to identify behaviourally similar ("neighbour") companies around each target (AAPL, GOOGL, NVDA) for target-specific cohort training. All features (18) were computed per ticker on date-sorted data, use rolling windows, and are standardised (winsorization + z-scores) to enable meaningful distance calculations.[1]

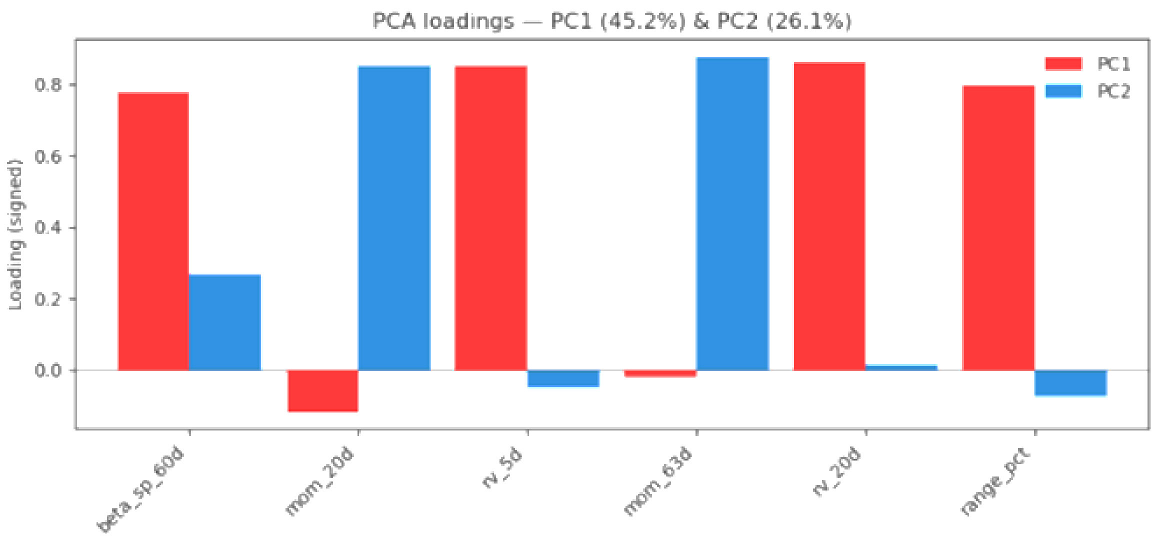
---

[1] Feature families included: Momentum (position & trend) (examples: RSI, Bollinger position, MACD 20-/63-day momentum sums); Volatility / Risk (examples: realised volatility (5/20-day), beta to S&P (60-day), idiosyncratic volatility, intraday range %); and Liquidity / Participation (Example: 60-day volume z-score).

Peer selection process included applying K-means clustering to two years of data (Aug 2023–Jul 2025) for all NASDAQ-100 members prior to distance-based selection. K-means clustering first partitioned the universe into two risk regimes (defensive /compounder vs. high-beta/ momentum). Nearest-neighbour cohorts were then selected within each target's regime using Euclidean distance between per-ticker centroids in the retained six-feature z-space, yielding stable peer sets for training. After filtering by business domain to retain only technology and communication companies, a fixed **top 5 neighbour cohort was set for each target:**

Google		Apple		Nvidia	
ADSK	Autodesk	**CDW**	CDW	**AVGO**	Broadcom
TXN	Texas Instruments	**MSFT**	Microsoft	**MRVL**	Marvell Technology
AMZN	Amazon	**CTSH**	Cognizant Technology Solutions	**MU**	Micron Technology
AAPL	Apple	**ADSK**	Autodesk	**AMD**	Advanced Micro Devices
ADI	Analog Devices	**GOOGL**	Google	**LRCX**	Lam Research





Box A.1: Data Overview

Date & Time Metrics

date	Day of the week	month	Days from month end	Current quarter	days to earnings	days since last earnings	earnings today (Y/N)

Google. Apple & NVIDIA + Neighbouring companies: Performance & Sentiment

Stock Prices: *Standard Daily Metrics* / *Relative P. Metrics*

Open – *Opening stock price*	**High –** *Day's highest price*	**Low –** *Day's lowest price*	**Close –** *Closing stock price*	**Volume –** *Shares traded that day*	**Price Movement -** *% change from previous close*	**Performance vs SP -** *Stock 5-day return – S&P 5-day return*	**S&P 500 5d return -** *(%)*

Momentum (Position & Trend)

Current price placement in short-term bands plus the magnitude and quality of the recent trend

RSI - *14-day RSI (Relative Strength Indicator)*	**Bollinger position –** *(Close – lower BB) ÷ (upper – lower)*	**SMA_ratio_20d –** *Close ÷ 20-day SMA*	**MACD –** *signal line (Moving Av. Convergence Divergence)*
mom_20d - *sum of last 20 daily returns*	**mom_63d -** *sum of last 63 daily returns*	**Sharpe_20d -** *mean(1-day returns, 20d) ÷ std(1-day returns, 20d)*	

Volatility / Risk

The typical magnitude, asymmetry, and co-movement with the market of price moves across single-day and multi-day horizons

rv_5d - *5-day realized vol: std(1-day returns, 5d)*	**rv_20d:** *20-day realized vol: std(1-day returns, 20d)*	**Bollinger_lower-** *20-day SMA + 2×std dev*	**Bollinger_upper -** *(Close – lower BB) ÷ (upper – lower)*
semivol_20d - *20-day downside vol: std(negative 1-day returns only, 20d)*	**dd_60d -** *60-day drawdown: close ÷ max(close, last 60d) – 1*	**range_pct –** *intraday range % - (high – low) ÷ close*	**gap_ret –** *overnight gap return - open ÷ prior close – 1*
beta_sp_60d - *60-day market beta: Cov(1-day stock, 1-day S&P, 60d) ÷ Var(1-day S&P, 60d)*		**idio_vol_20d - 20-day** *idiosyncratic vol: std(stock return – β_{60} × market return , 20d)*	

Liquidity / Participation

The unusualness of trading volume versus the stock's own history, summarizing participation intensity

vol_z_60d - *60-day volume z-score: (volume – mean(volume, 60d)) ÷ std(volume, 60d)*

Earnings Data: Key Metrics

EPS Actual	**EPS Expected**	**EPS surprise –** *(Actual – Expected EPS) ÷ Expected EPS*	**Quarterly Revenue** *(Actual Revenue)*

Financial News & Earnings Call Transcripts – Sentiment Scores

Company's Sentiment	**AI Sentiment**	**Tariffs-Related**	Other

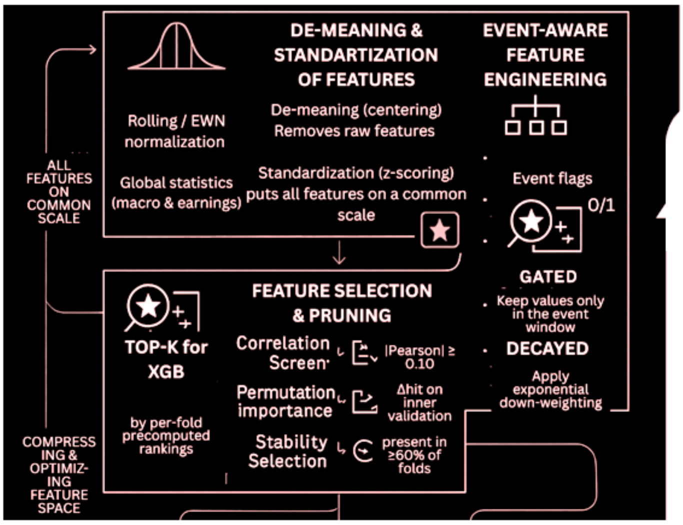
Macroeconomic Factors: Mixed-frequency macro matrix (no leakage)

Interest Rate	**Inflation (CPI, MoM, YoY)**	**Unemployment Rate**	**Trade (balance, imports, exports)**	**GDP**	**Household Debt Service Ratio**

Code Overview

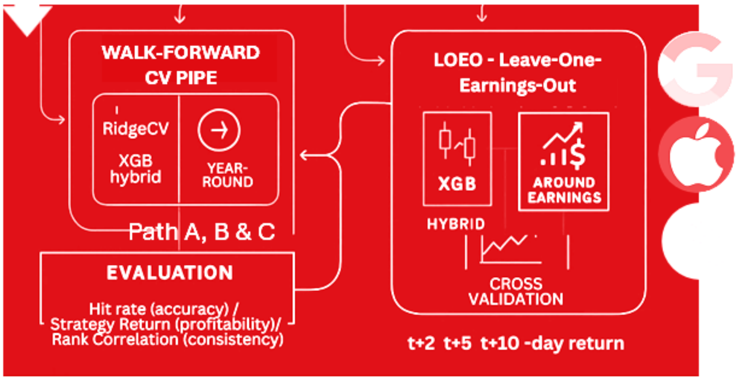
Technical overview of the Python pipeline

The predictive modeling pipeline comprises (a) *pre-modelling transformations* that de-bias, standardise, and prune the feature space to prevent mixed, redundant signals and noise, and (b*) two ready-to-run modelling paths*: Walk-Forward Cross-Validation (WFCV) for year-round predictions and Leave-One-Earnings-Out (LOEO) for earnings-window predictions. [2]



Pre-modelling
Data transformation & smart filtering

Modelling Paths
Year-round & around-earnings pred.



Below is a brief code overview of the main pipeline, documenting key metrics and parameters to ensure reproducibility and facilitate future refinement.

Pre-Modelling Transformations

De-Meaning and Standardisation of Features

Raw market and macro features exhibit drifts, seasonality, and heterogeneous scales—risking spurious trend learning and scale bias. To mitigate these risks, family-specific de-meaning and standardization were applied using rolling windows, EWM, and global statistics (see *Box A.2: De-meaning and Standardization of Features* – for full methodological documentation).

---

[2] See Appendix for full methodological notes.

Normalization & De-Meaning Methods by Feature Family

De-Meaning & Normalization Method	Feature Family	Features Included	Outputs
	Price levels	*open_*, high_*, low_*, close_**	Raw + *_dm, *_z
Rolling mean/ std (60d)	Distance-to-trend / Momentum	*bollinger_position_*, macd_*, rsi_*, sma_ratio_20_*, ret_1d_*, ret_5d_*, perf_vs_sp_ret_5d_*, rolling_avg_10d_*, mom_20d_*, mom_63d_*, sharp_20d_*, daily_pct_change_**	Raw + *_dm, *_z
	Market returns	*sp_ret_1d, sp_ret_5d*	Raw + *_dm, *_z
	Sentiment	*sentiment_* ai_score_**	Raw + *_dm, *_z
	Volume / Turnover	*volume_*, vol_z_60d_**	Raw + *_dm, *_z
EWM mean / std (half-life 25d)	Volatility / Beta	*atr_*, beta_sp_60d_*, bollinger_upper_*, bollinger_lower_*, rv_20d_*, semivol_20d_*, idio_vol_20d_*, dd_60d_**	Raw + *_dm, *_z
Global mean / std (full sample)	Macroeconomic levels	*cpi_raw, cpi_mom, cpi_yoy, exports_raw, exports_mom, exports_yoy, imports_raw, imports_mom, imports_yoy, trade_balance_raw, trade_balance_mom, trade_balance_yoy, unemp_rate, gdp, debt_service_ratio, interest_rate_daily*	Raw + *_gz
	Earnings levels	*reported_eps_*, estimated_eps_*, revenue_million_*, surprise_percent_*, earning_date_*, earnings_date_*, earning_date_actual_*, quarter_ending_**	Raw + *_gz

Code Snippet: Trailing Rolling Mean & EWM

```python
# --- helpers: leak-safe rolling / EWMA stats ---
# Compute trailing rolling mean (mu) and standard deviation (sigma) for each column
# in `block`, then shift by 1 day to avoid look-ahead bias.
def _rolling_mu_sigma(block: pd.DataFrame, window: int, minp: int | None = None):
    if minp is None:
        minp = min(25, max(1, window // 2))
    mu = block.rolling(window=window, min_periods=minp).mean().shift(1)
    sigma = block.rolling(window=window, min_periods=minp).std(ddof=0).shift(1)
    return mu, sigma

# Compute exponentially weighted mean (mu) and standard deviation (sigma)
# with a specified half-life, then shift by 1 day to avoid look-ahead bias.
def _ewm_mu_sigma(block: pd.DataFrame, halflife: int, minp: int = 10):
    # adjust=False is standard for online filtering; shift(1) to avoid leakage
    mu = block.ewm(halflife=halflife, adjust=False, min_periods=minp).mean().shift(1)
    # bias=False → sample std analogue; min_periods to avoid early junk
    sigma = block.ewm(halflife=halflife, adjust=False, min_periods=minp).std(bias=False).shift(1)
    return mu, sigma

# adding for unique standartization of low frequency features
def _global_mu_sigma(block: pd.DataFrame):
    # one mean/std per column over the whole sample; broadcast to all rows
    mu = pd.DataFrame({c: block[c].mean() for c in block.columns}, index=block.index)
    std = pd.DataFrame({c: block[c].std(ddof=0) for c in block.columns}, index=block.index)
    return mu, std
```

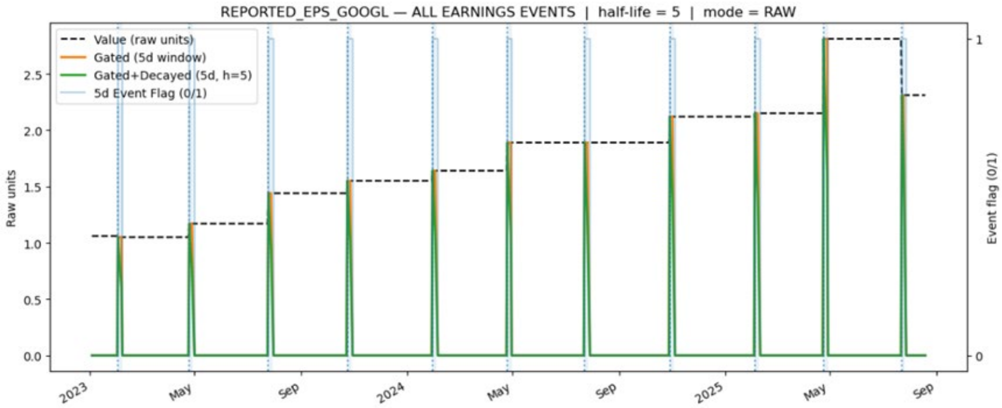
These methods yield stationary, volatility-adjusted features that enhance comparability, purify signal, and improve predictive robustness by focusing on relative deviations across uniform scales.

Event-Aware Feature Engineering

Low-frequency releases (macro and earnings) – which risk spurious trend learning and dilution of impact if simply forward-filled – were transformed to reflect their time-localized effect by engineering event-aware features: *event flags* (0/1; flag ±W business-day windows around anchors); *gated*, *decayed* and combined *gated+decayed* features (with half-life tuned to horizon – macro: ~7–20 days; earnings: ~5 days).

Code Snippet: Gated + Decayed Features

```python
# Compute exponential decay weights from a "days since release" series.
# - If half_life is None → return None (no decay).
# - Otherwise: clip days to [0, cap_days] and apply 0.5 ** (days / half_life).
def _decay_weights(since_series: pd.Series, half_life: int | None, cap_days: int) -> pd.Series | None:
    if half_life is None:
        return None
    d = since_series.clip(lower=0, upper=cap_days).astype(float)
    return np.power(0.5, d / float(half_life))
```



Event-aware features eliminated spurious persistence and trend leakage while elevating performance by directing capacity to authentic release-time signals (see *Box A.3: Event-Aware Feature Engineering*– for full methodological documentation).

Feature Selection & Pruning

To mitigate variance inflation from a broad signal set, a robust filtering and pruning sequence included: *deterministic prefilter* (standardized-over-raw; correlation screen); Leakage-safe *PI* on inner validation (Δhit threshold ≈ 1pp); *ridge stability selection* (retain if present in ≥60% of folds); and Ridge-ordered ***Top-K** funnel* for XGB.

Code Snippet: Ridge Stability Selection

```python
def ridge_stability_select(dfm, splits, target_col, K=40, min_frac=0.6):
    """
    Across folds, fit Ridge on TRAIN and record the top-K |coef| features.
    Keep features that appear in >= min_frac of folds.
    """
    from sklearn.linear_model import RidgeCV

    X = dfm.drop(columns=[target_col])
    y = dfm[target_col].values
    cols = np.array(X.columns)

    counts = pd.Series(0, index=cols)
    for tr_idx, te_idx in splits:
        mdl = RidgeCV(alphas=np.logspace(-4, 3, 20)).fit(X.iloc[tr_idx].values, y[tr_idx])
        top = cols[np.argsort(np.abs(mdl.coef_))[::-1][:K]]
        counts[top] += 1

    keep = counts[counts >= min_frac * len(splits)].index.tolist()
    return keep, counts.sort_values(ascending=False)
```

Code Snippet: Permutation Importance

```
def perm_importance_by_fold_val(model_fn, dfm, splits, target_col, feat_cols=None, n_repeats=30):
    """
    For each (train, test) fold:
      - time-split TRAIN into (subtrain, val)
      - fit model on subtrain
      - compute permutation importance on val (no leakage from outer test)
    Returns a DataFrame with per-fold importances and their mean.
    """
    X_all = dfm.drop(columns=[target_col])
    if feat_cols is None:
        feat_cols = list(X_all.columns)
```

NOTE: See Box A.4: Selection of Features and Pruning Methods – for full methodological documentation).

Modelling Paths: WFCV & LOEO

Walk-Forward Cross-Validation

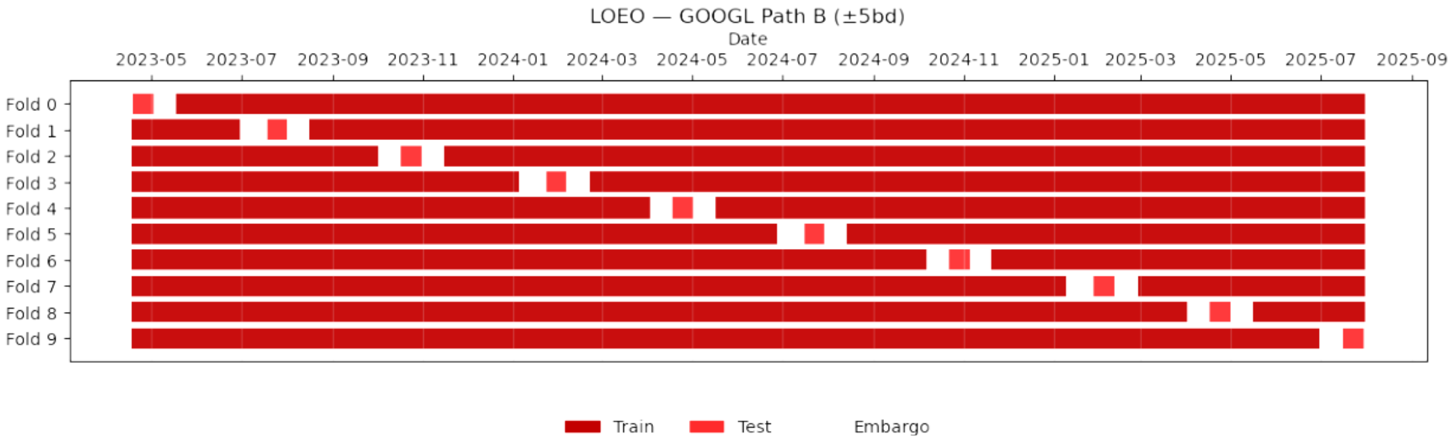
WFCV replaces a single random end-split with a sequence of chronological experiments, each training on ~1 trading year, testing on ~1 quarter (~63 trading days), and advancing by ~1 month. An embargo equal to the horizon filters rows whose forward returns would overlap test labels (see *Box A.5:Walk-Forward Cross-Validation*– for full methodological documentation).

Code Snippet: Walk-Forward Expanding Folds

```
def make_walk_forward_splits(dates, train_min=252, test_size=63, step=21, purge=10):
    """
    dates: pd.Series or index of datetime
    train_min: min training length (e.g., 1Y of trading days)
    test_size: length of each test block (e.g., 3 months ~ 63 trading days)
    step: how far we slide the window each fold
    purge: drop +/- 'purge' days around test from train (for T+10 targets use purge=10)
    """
    idx = np.arange(len(dates))
    out = []
    start = train_min
    while start + test_size <= len(dates):
        train_end = start  # exclusive
        test_start = train_end
        test_end = test_start + test_size

        tr = idx[:train_end]
        te = idx[test_start:test_end]

        # purge: drop train samples with date within [test_start - purge, test_end + purge]
        lo = max(0, test_start - purge)
        hi = min(len(idx), test_end + purge)
        keep_train = np.setdiff1d(tr, idx[lo:hi], assume_unique=True)

        out.append((keep_train, te))
        start += step
    return out
```

Leave-One-Earnings-Out (LOEO)

LOEO retains the CV logic but shifts the validation unit from calendar-based slices to discrete earnings events. For each fold, one earnings anchor (day 0) is held out. The test window is ±W business days (e.g., W ∈ {2, 5, 10}); an embargo of H days (the prediction horizon) is applied on both sides. The train set comprises all dates outside [A−W−H, A+W+H] (see *Box A.5: Leave-One-Earnings-Out (LOEO)* – for full methodological documentation).



Evaluation Metrics

Both WFCV and LOEO were set to report directional accuracy, Spearman's ρ, and conditional strategy returns per fold (expanding /earnings) and in aggregate (cross-fold mean with variability), with the most recent fold highlighted.

Code Snippet: Evaluation Metrics

```
def eval_metrics(y_true, y_pred):
    """
    Computes three simple, interpretable metrics per fold:
      - hit: directional accuracy, i.e., % of times sign(y_pred) == sign(y_true)
      - strat_ret: average realized return under a naive rule:
                   go long if prediction > 0, else stay flat.
                   (For cross-sectional portfolios, replace with long-short spread.)
      - spearman: rank correlation (Spearman) between predictions and realized returns,
                  robust to monotonic transformations of the score.
    """
    # Directional accuracy (classification-style correctness on the sign)
    hit = (np.sign(y_pred) == np.sign(y_true)).mean()
    # Simple strategy return: average realized return on the subset where the model is long
    strat_ret = y_true[y_pred > 0].mean() if np.any(y_pred > 0) else 0.0
    # Rank correlation (monotonic association); NaNs ignored by 'omit'
    rho = spearmanr(y_true, y_pred, nan_policy="omit").correlation
    return {"hit": hit, "strat_ret": strat_ret, "spearman": rho}
```

They evaluate generalization *across regimes* and offer a time-robust long-run view with a current-regime benchmark. In both cases, the pipeline was configured to allow swapping models, paths, and horizons to test and select the best-performing candidate.

Code Snippet: WFCV & LOEO – Ready-to-Run Pipes

```
# ==========================================================================
# Final, prunned, mocro and earnings tied PATH B (TARGET-ONLY), PATH A (PEER-ONLY) and PATH C (TARGET + PEERS) SETUP + BASELINE EVALUATION
# ==========================================================================
# --------------------- CONFIGURATION ---------------------------------------
target_ticker = "googl"     # Target ticker (lower-case to match column name suffixes): Insert aapl for APPL and nvda for NVDA
H = 10                      # Evaluation horizon in trading days (e.g., 2, 5, 10, 20)
target_col = f"future_{H}d_ret_{target_ticker}"  # e.g., "future_10d_ret_googl"
# Select the augmented frame for the target ticker.
df_aug_tied = wide_google_aug_tied    # Insert wide_apple_aug_tied for AAPL, wide_nvda_aug_tied for NVDA
df_aug_tied = prepare_master_for_corr(df_aug_tied) # turn quarter, day of the week, into numeric
peer_sets = {
    "googl": ['GOOGL','ADSK','TXN','AMZN','AAPL','ADI'],
    "aapl":  ['AAPL','CDW','MSFT','CTSH','ADSK','GOOGL'],
    "nvda":  ['NVDA','AVGO','MRVL','MU','AMD','LRCX'],
```

WFCV / Year-Round Pred.

```
df_wide = wide_map["googl"]       # or pick_wide("googl") chnage to nvda or apple as needed
H       = 10                      # Choose target:  t+_ days return
path    = "B"                     # Choose path: "A" | "B" | "C"

loeo = run_loeo_full(
    df_wide=df_wide,
    target_ticker="googl",
    H=H,
    path=path,
    window=5,               # ±5 business days before and after earnings - Choose Window (=2, =5, =10)
    purge=H,                # embargo = horizon
    Ks=(20,30,40,60),
    final_model="xgb",
    verbose=True,
    do_shap=False,
    peer_sets=peer_sets     # <-- required for path "B"
)
```

LOEO / Pred. Around Earnings

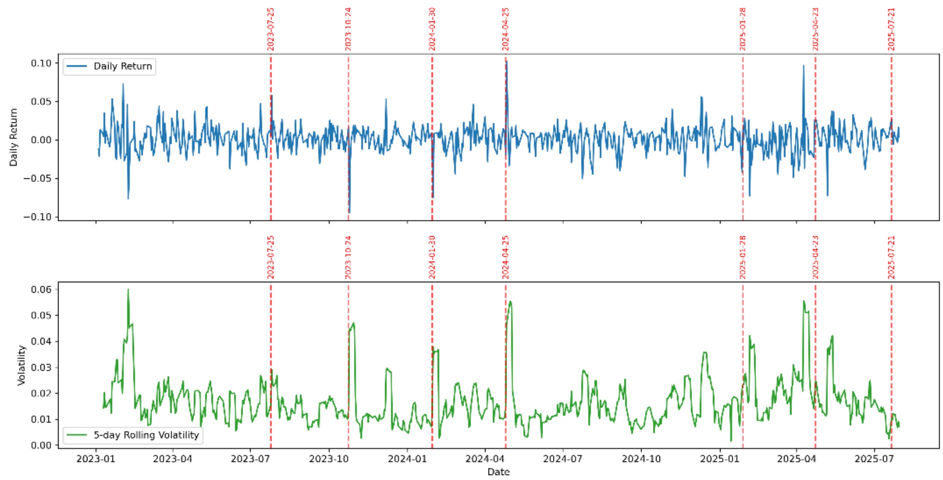
Patterns and insights

Key findings from exploratory analysis, model development, and strategy implications

Pre-Modeling Explorations: volatility metrics & predictive modeling

Volatility measures the speed and magnitude of price changes in securities, with high volatility indicating greater risk and unpredictability. Earnings announcements often cause sharp volatility spikes due to shifts in investor sentiment and market expectations.

GOOGL Daily Returns and Volatility with Earnings Days



Incorporating volatility metrics around earnings into predictive models is expected to improve short-term stock forecasting by capturing dynamic price behaviour.

Volatility by company

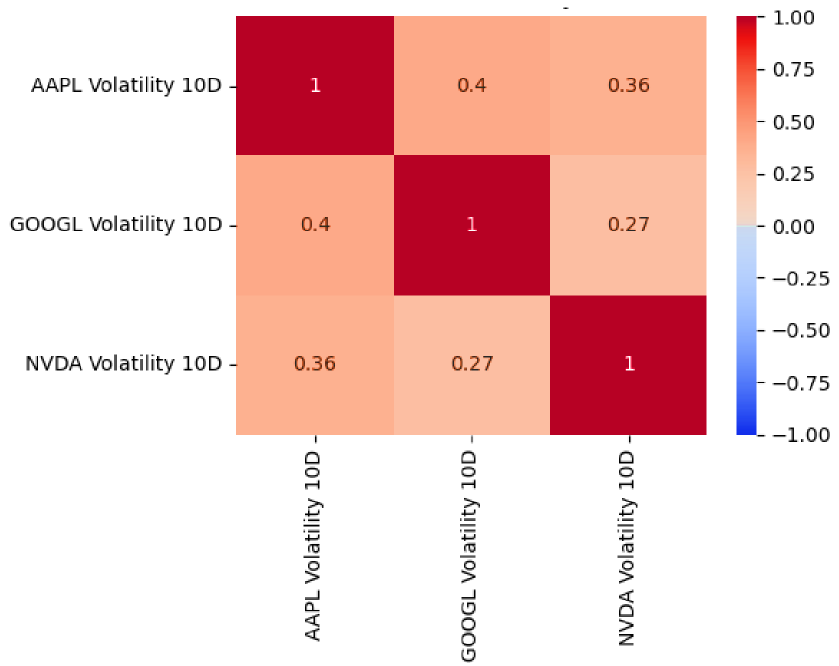
Analysis of **10-day return volatility decomposition** reveals that Nvidia exhibits the highest overall volatility, with notable spikes in mid-2023 and late 2024, signalling intense market reactions and a clear upward trend in uncertainty. Apple shows moderate volatility that gradually increases, particularly in 2025. In contrast, Google maintains relatively low and stable volatility with only mild fluctuations, indicating more consistent price behaviour.

Volatility & Bollinger predictive potential by horizons

Bollinger Bands helped identify potential buy and sell signals. However, the effectiveness of these signals varies with the time window. Across Apple, Google, and Nvidia, short-term (2-day) price movements are highly volatile and unpredictable, reducing the reliability of technical indicators. At the 5-day level, clearer patterns start to emerge, making Bollinger Band signals more useful for swing trading. By the 10-day horizon, volatility stabilises, and the bands become more effective for forecasting returns. These stocks tend to move in the **upwards movement**, exhibiting **moderate positive correlations** in returns over 2-, 5-, and 10-day periods.

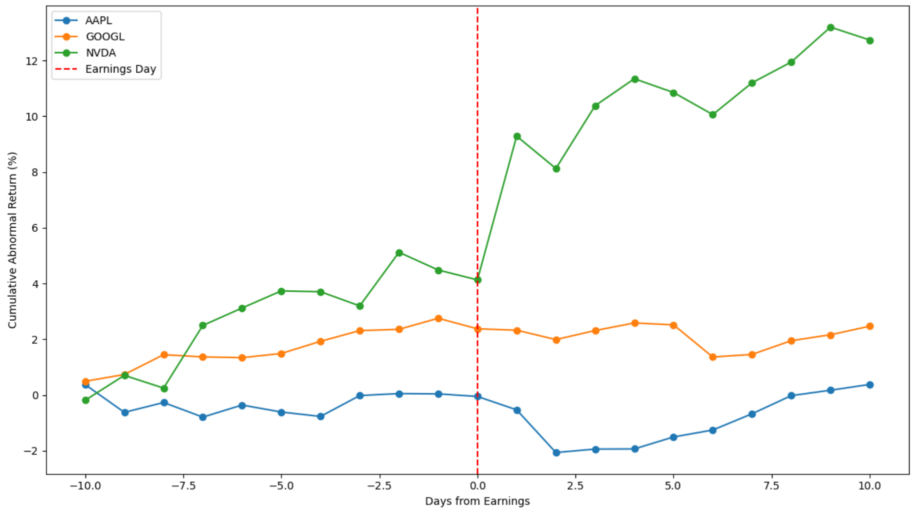
Apple and Google share the strongest return correlation, likely due to their shared status as large-cap tech companies affected by similar macroeconomic and sectoral trends. **Nvidia**, while still positively correlated, shows weaker co-movement due to its higher volatility and exposure to different industry dynamics. In contrast, **volatility correlations** among these stocks are generally **weaker than their return correlations**, meaning their price directions often align more than their risk levels do.

Correlation Matrix of 10-Day Volatilities



Weaker Cross-Firm Volatility Correlations Signal Divergent Risk Profiles.

Cumulative Abnormal Return (CAR) Around Earnings



Post-earnings analysis of Apple, Google, and Nvidia reveals significant differences in return behaviour and volatility**.**

Nvidia consistently shows the strongest and most volatile post-earnings reaction, evidenced by a sharp and sustained increase in **Cumulative Abnormal Returns (CAR)** and a wide spread of returns with high positive outliers. This indicates strong investor response and heightened market volatility, likely driven by NVDA's growth-focused business and exposure to sectors like AI and semiconductors.

Google exhibits a moderate post-earnings reaction, with increased return dispersion and both positive and negative outliers, reflecting mixed investor sentiment. AAPL remains the most stable, showing only slight increases in CAR and minimal changes in return distribution.

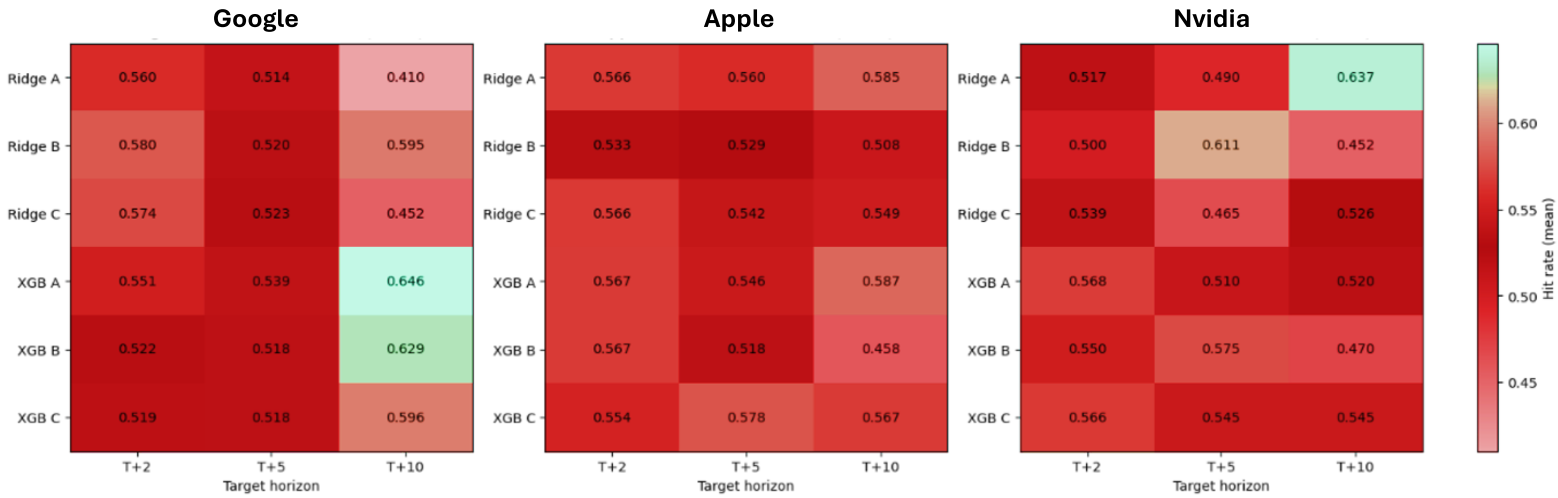
The best performing model

Walk-Forward Cross-Validation

Late-period strength, regime-robust results, overlay alignment, and balanced SHAP drivers make the case for XGB Hybrid Path B (Google, T+10) as the near-term deployment choice.

WFCV: Hybrid XGB (Path B) proved most effective for Google at T+10[3]

Year-Round Ridge+XGB Model Hit Rates by Company, Path, and Horizon



NOTE: Relative to Path A, Path B exhibits more volatility in early folds but outperforms in later folds; For NVIDIA (T+10), Ridge performance was relatively high, but given that MLR served solely as the baseline comparator, this target was not pursued further in the current project.

XGB Hybrid – Path B (Target Company)

fold	hit	spearman	strat_ret
0	0.73	-0.008	3.401
1	0.556	-0.011	-0.342
2	0.444	-0.293	-2.063
3	0.492	0.067	-1.784
4	0.619	0.139	0.345
5	0.619	0.077	1.019
6	0.667	-0.119	2.773
7	0.683	0.376	2.73
8	0.556	0.182	0.691
9	0.603	0.078	-2.892
10	0.746	0.43	1.422
11	0.651	0.391	4.198
12	0.683	-0.093	2.646
13	0.762	0.016	3.52

Means: hit=0.629, rho=0.088, ret=1.1189

Directional Accuracy, Rank Alignment, and Conditional Returns by Fold – Showing the XGB Path B as the Most Time-Robust, High-Performance Model

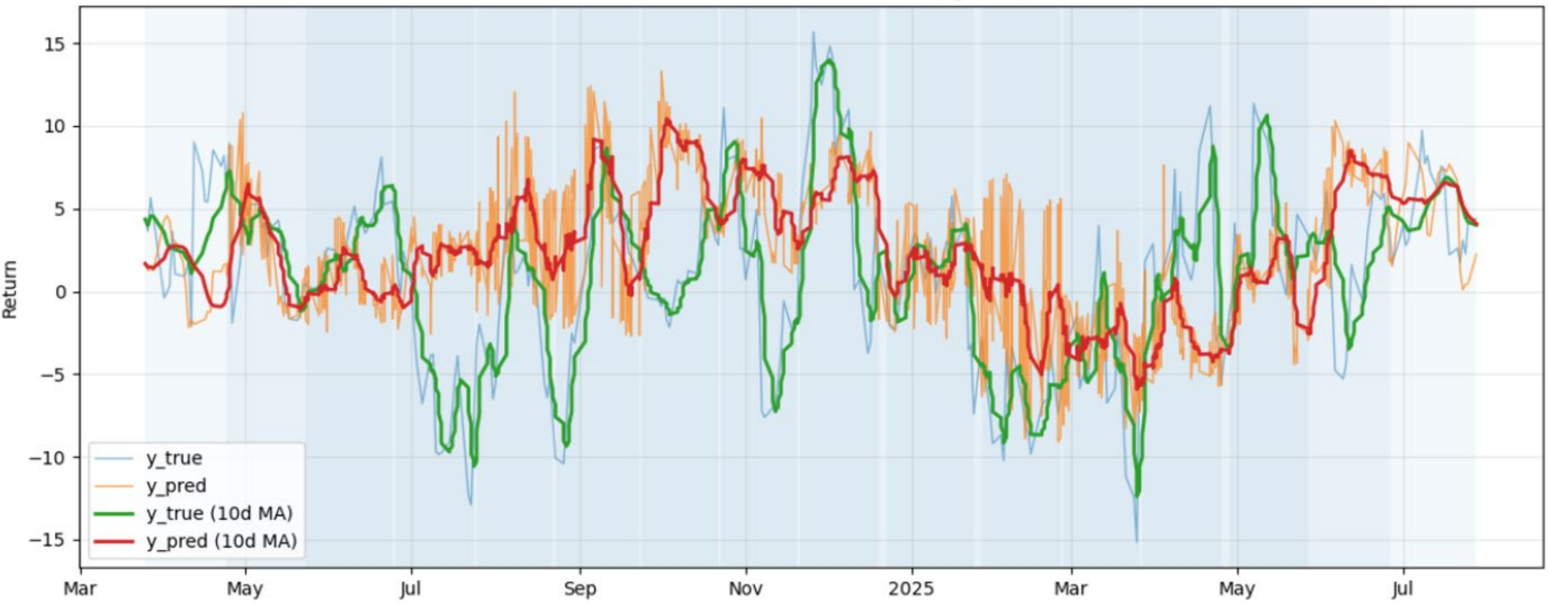
Under the WFCV protocol, the Hybrid XGB model (Path B, Horizon t+10) demonstrates **strong and improving performance through time.** Directional accuracy is consistently ≥0.60 in later folds, with the most recent fold delivering the highest results. Spearman's ρ strengthens toward the end of the sample, indicating better rank alignment, and conditional long-only returns are positive in the majority of folds. Taken together, the late-period metrics and payoff profile indicate robust generalization and near-term deployability.

[3] For NVIDIA (T+10), Ridge performance was relatively high, but given that MLR served solely as the baseline comparator, this target was not pursued further in the current project.

Later folds show closer tracking and improved directional alignment

Prediction series converging toward the realized path in later periods, with the 10-day moving averages nearly overlapping during several swings.

Google's Actual vs. Predicted T+10-Day Return Over Time (by Fold)



.

	mean|SHAP|
beta_sp_60d_googl_z	1.667461
trade_balance_yoy_gz	1.350686
estimated_eps_googl_gz__decay_h5	1.343937
cpi_yoy__decay_h7	0.488327
imports_yoy_gz	0.330009
cpi_mom__gate_event_5d	0.202808
debt_service_ratio_gz__gate_dsr_event_5d_dec_h20	0.177304
revenue_million_googl_gz	0.110802
surprise_percent_googl_gz	0.091830
reported_eps_googl_gz__gate_earnings_date_5d_googl_dec_h5	0.086347

SHAP importance: balanced mix of market beta, macro trade/CPI, and earnings decay/gates.

SHAP analysis for the last fold attributes importance to a balanced set of drivers: ***market beta*** (beta_sp_60d_googl_z), ***external trade and macro signals*** (e.g., trade_balance_yoy_gz, cpi_yoy_decay_h7, imports_yoy_gz), and ***earnings constructs*** with event gating/decay (e.g.,estimated_eps _googl_gz_decay_h5, reported_eps_googl_ gz_gate_earnings_date _5d_dec_ h5). This composition is consistent with a model that captures both broad market conditions and time-localized earnings effects rather than overfitting to a single signal family.

Leave-One-Earnings-Out (LOEO)

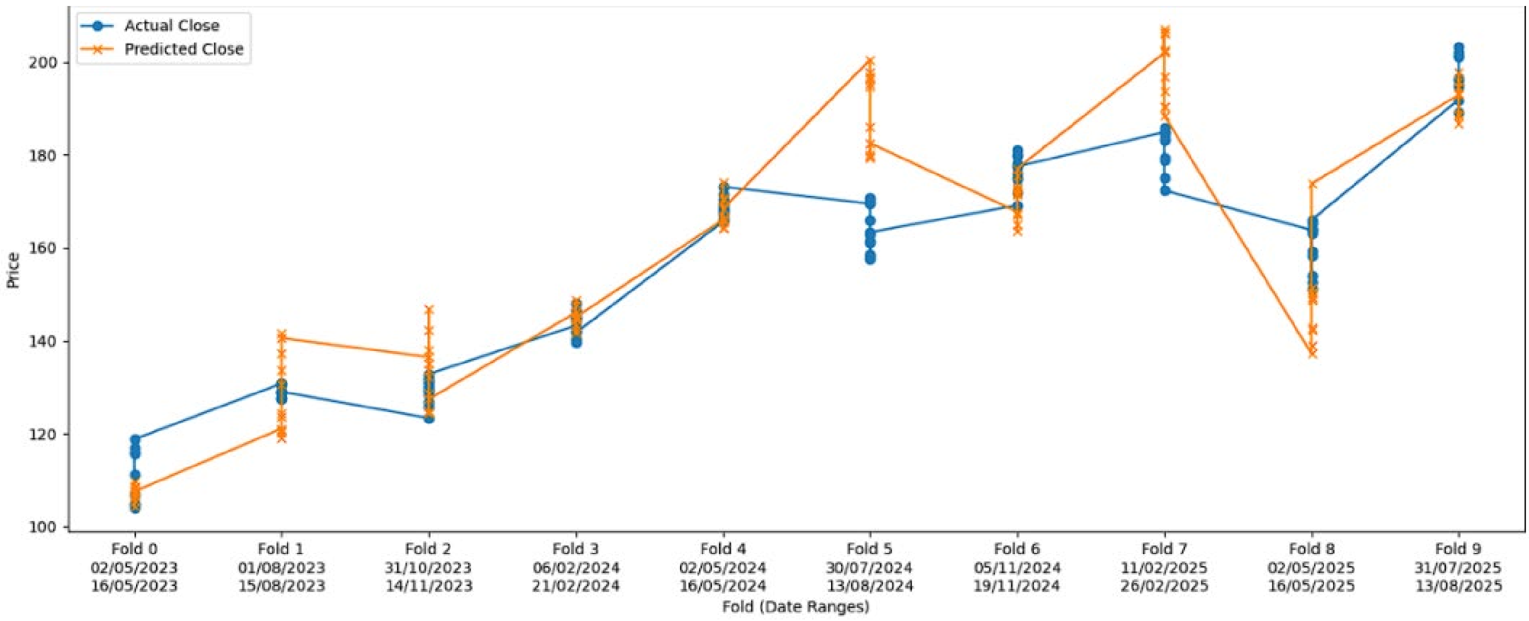
Under the LOEO regime (10 aligned earnings events), the Google T+10 model delivered consistently strong hit rates around earnings, with several folds in the 0.70–0.90 range and a last-fold hit ≈0.64; the cross-fold mean hit ≈0.59 reflects event-to-event variability typical of earnings windows. Rank alignment is modestly positive on average and conditional long-only returns are positive on balance. Feature inventories for the best configuration combine market beta with event-aware macro and earnings signals (gates/decays), supporting the view that time-localized release effects drive much of the predictive lift.

Mean Directional Hit Across Earnings Events, t+10-day Returns

Fold	Google	Apple	Nvidia
0	0.818	0.545	(-)
1	0.545	0.091	0.7
2	0.727	0.182	0.273
3	0.727	0.636	0
4	0.909	0.545	1
5	0	0.636	0
6	1	0.545	0.3
7	0	0.727	0.273
8	5	0.364	0.182
Last (9)	**0.636**	**0.889**	1
Average	**0.59**	**0.52**	**0.41**

Actual vs. Predicted T+10 Close shows clear alignment, indicating useful directional information even when exact price levels differ—supporting the use of forecasts as trading signals rather than point-price targets.

Actual vs. Predicted Close Price (T+10)



NOTE: Root Mean Squared Error (RMSE): 12.78; **R^2 Score: 0.76**

Investment & Trading Strategies

Year-Round Google Trading Strategy

Strategy specification: signals are the model's 10-day return forecasts (XGB Hybrid, Path B). A quantile long/short rule is applied each day:

- ➔ **Go long** when the forecast is in the top 15% of its in-sample distribution;
- ➔ **Go short** when it is in the bottom 10%;

Otherwise, hold no position. Positions are dynamically sized with a decaying weight ladder so newer signals receive larger weights than older ones. Per-trade exposure is capped at 5% of capital to limit concentration. Transaction costs of 3 bps per trade are debited at execution.

Risk targeting and comparability: to compare fairly with buy-and-hold, the strategy is *volatility-scaled to 10% annualised*; a scale factor of 7.28 (computed on ex-ante risk) is applied to the signal to hit the target risk.

With Costs and Position Caps: Forecast-Driven Strategy Outperforms Buy-and-Hold at Equal Volatility

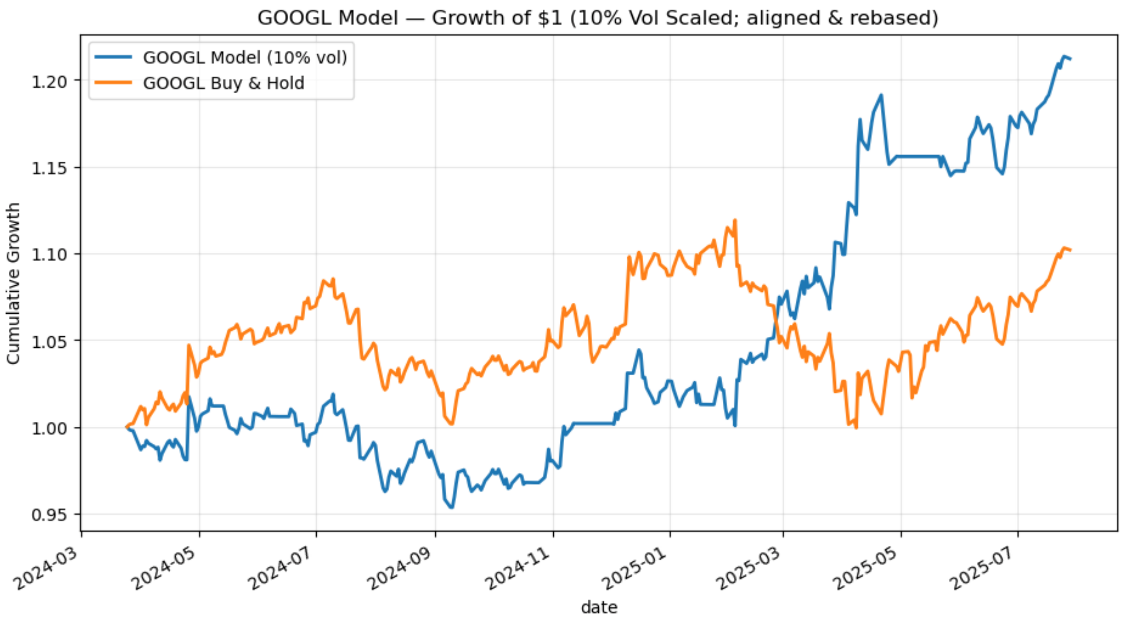


Fig – Year-Round Google Trading Model vs Buy-and-hold strategy (10% annualised volatility)

Strategy	Annual Return	Annual Volatility	Sharpe Ratio	Max Drawdown	Scale Factor	Hit Ratio	Spearman IC
GOOGL Model 85/10	**15.52%**	**10.00%**	**1.49**	**-6.41%**	**7.28**	**66.4%**	**0.195**
GOOGL Buy & Hold (10% vol)	7.44%	10.00%	0.77	-10.70%	0.33	n/a	n/a

Results: In the "Growth of $1" chart (10% vol, aligned & rebased), the model-driven curve compounds faster and with smoother drawdowns than buy-and-hold. The summary table shows that **with costs and position caps, the strategy converts forecasts into superior, risk-matched performance and a smoother equity curve than buy-and-hold.**

Earnings Window Investment Strategy

Strategy definition: the model's daily T+10 return forecast determines position state.

At the start of each walk-forward fold:

- ➔ **Go long** on the first strictly positive forecast.
- ➔ **Hold** position for 10 trading days, after which:

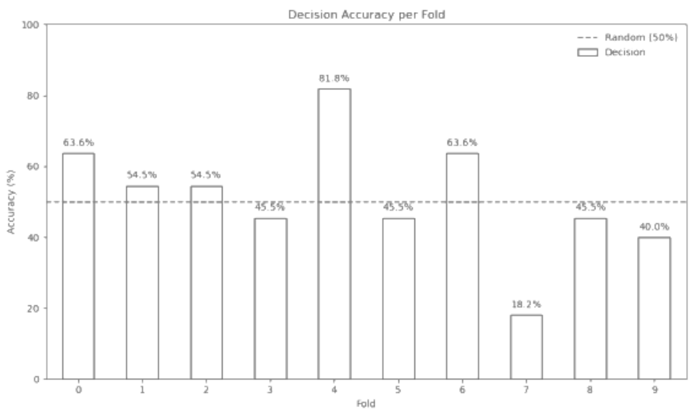
  The rule is evaluated each day:

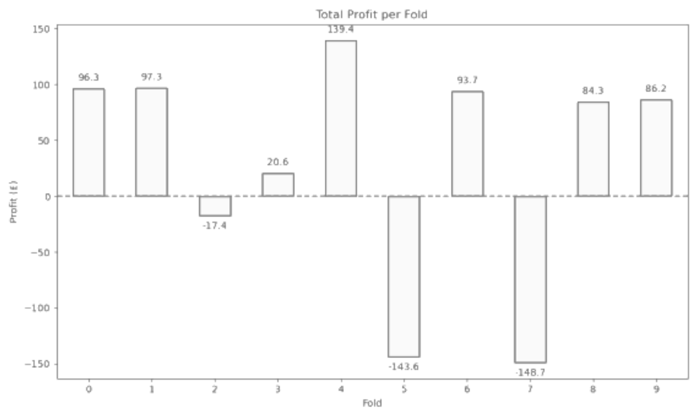
  - *Maintain* the long if the T+10 forecast remains positive;
  - *Exit* (go flat) if the forecast turns negative.

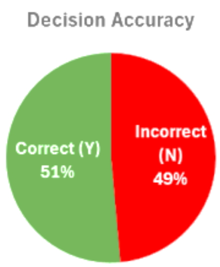
  The cycle repeats until the fold ends, at which point any open position is closed. No leverage or short exposure is used in this variant.

This rule operationalizes the predictive signal in a conservative, long-only manner: it monetizes direction where confidence is positive and stands aside when it is not. The evidence – moderate accuracy paired with positive cumulative P&L – suggests the forecast contains tradable information, though performance is regime-dependent.

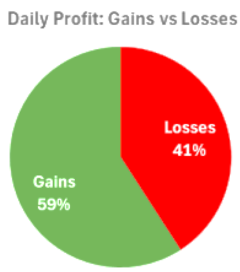
T+10 Buy/Hold/Sell Strategy – Accuracy and Profitability by Fold: Aggregate gain 30.8% with 51% correct decisions across folds.







The 6 Decision Outcomes		
Buy	Price Goes Up	✅ Good
	Price Goes Down	❌ Bad
Hold	Price Goes Up	✅ Good
	Price Goes Down	❌ Bad
Sell	Price Goes Down	✅ Good
	Price Goes Up	❌ Bad



NOTE: The left panel shows fold-level decision accuracy (benchmark line at 50% "coin-flip"). The right panel reports total profit per fold, highlighting that profitability is concentrated in several folds despite near-even overall hit rates – i.e., gains are asymmetric when the model is right. The pie charts summarize 51% correct vs. 49% incorrect decisions and 59% gain days vs. 41% loss days.

Results: Across folds, the approach achieves an **aggregate gain of ~30.8%** with **51% decision accuracy** (share of daily decisions that improved P&L relative to the alternative action). Results vary by fold – several periods contribute outsized profits, while a few show drawdowns—consistent with regime effects.

Pair Trade Strategy

Pair trading is a short-term strategy that profits from temporary price divergences between two historically correlated stocks. Using our dataset of 15 peer stocks, cointegration tests (p-value < 0.05) identified GOOGL and AMZN to be the only correlated pair.

Testing cointegration for peers:

Base Stock	Pair Stock	p-value
AAPL	CDW	0.3501
	MSFT	0.5669
	CTSH	0.2237
	ADSK	0.1150
GOOGL	ADSK	0.3259
	TXN	0.3095
	AMZN	0.0089
	AAPL	0.2254
	ADI	0.0893
NVDA	AVGO	0.3004
	MRVL	0.9296
	MU	0.7011
	AMD	0.9725
	LRCX	0.8319

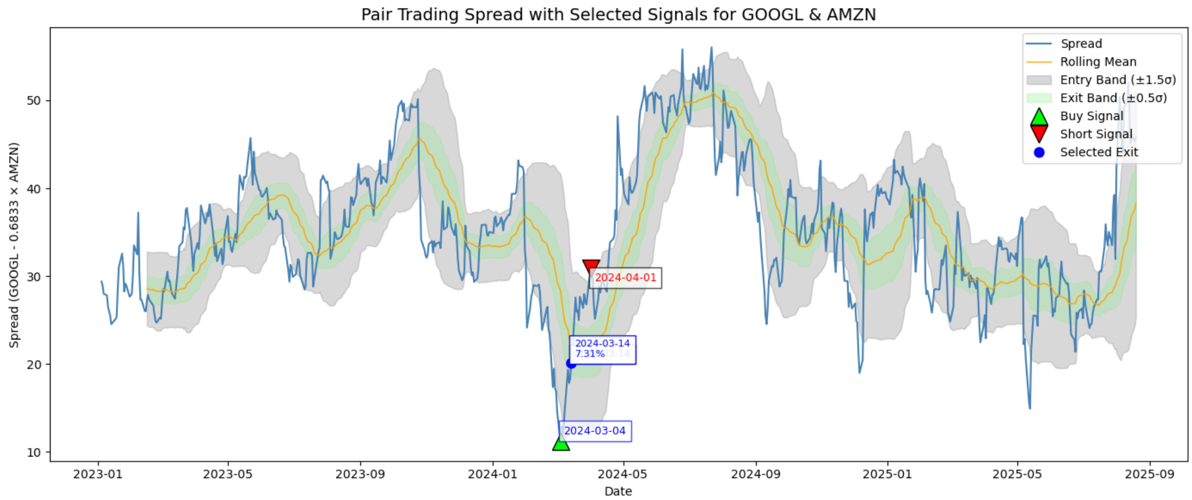
NOTE: For the pair, the hedge ratio (β = 0.6833) was estimated using OLS regression, and the spread was calculated as:

Spread = PriceGOOGL − β × PriceAMZN

The z-score of the spread was used to standardize deviations.

Trades are triggered when the z-score exceeds ±1.5 and exited when it reverts within ±0.5. (*i.e, for z > 1.5, short GOOGL and long AMZN; for z < -1.5, do the opposite*)

Trading Strategy: Begin trading upon the first long (buy) signal, close the position at the first exit signal, and initiate a short position at the first short signal.



Trading Strategy Rules Based on Spread Z-Score

Z-Score Range	Action	Meaning
spread_z_score > +1.5	▽ Short Entry	Spread is overvalued
spread_z_score < -1.5	△ Long Entry	Spread is undervalued
-0.5 < Z < +0.5	○ Exit	Spread near mean (take profit or close position)
±0.5 ≤ Z ≤ ±1.5	⊘ Hold / No Trade	Wait for a stronger signal

Incorporating Sentiment

To evaluate the relationship between sentiment and returns, two correlation metrics were examined:

(A) **Spearman IC** (rank correlation with future returns), capturing the monotone association of feature magnitudes with the target; and

(B) **Hit rate** (sign agreement), measuring the fraction of cases where sign(feature) == sign(target) – computed both ***year -round*** and ***within earnings windows*** (±5 business days) showed that **within earnings windows** (±5 business days), sentiment features display a marked improvement in predictive power.

Signal Checks Confirm: AI-Sentiment Score Matters for Predictions Around Earning

Spearman IC and Hit Rate of Sentiment Features vs. Google's T+10-Day Return,
Compared Outside and Inside Earnings Event Windows

GOOGL – overall:

	feature	n	IC_spearman	hit_rate
0	total_confidence_score_z	602	0.116	0.535
1	weighted_avg_sentiment_raw	642	-0.070	0.519
2	avg_confidence_score_raw	642	0.068	0.631
3	avg_sentiment_score_raw	642	-0.065	0.525
4	company_score_raw	646	-0.054	0.500
5	ai_score_z	606	0.045	0.533
6	ai_score_dm	606	0.044	0.533
7	ai_score_raw	646	-0.043	0.627
8	avg_confidence_score_dm	602	0.024	0.515
9	avg_confidence_score_z	602	0.023	0.515
10	weighted_avg_sentiment_dm	602	-0.023	0.528

GOOGL – inside earnings windows (±5bd):

	feature	n	IC_spearman	hit_rate
0	ai_score_dm	109	0.177	0.569
1	avg_confidence_score_dm	107	0.164	0.523
2	ai_score_z	109	0.160	0.569
3	avg_confidence_score_z	107	0.158	0.523
4	total_confidence_score_z	107	0.143	0.589
5	avg_confidence_score_raw	118	0.125	0.517
6	ai_score_raw	120	0.109	0.508
7	avg_sentiment_score_raw	118	0.081	0.508
8	weighted_avg_sentiment_raw	118	0.067	0.508
9	company_score_raw	120	-0.065	0.550
10	company_score_z	109	-0.044	0.578

Within earnings windows the de-meaned AI sentiment score (ai_score_dm) and its standardized counterpart (ai_score_z) lead with **meaningful rank correlations against the target** (IC ≈ 0.177 and 0.160) and are more often than not aligned with the target's sign (hit ≈ 0.569 and 0.589).

Despite encouraging signal checks around earnings, incorporating sentiment features hurt modeling performance in both regimes – year-round (WFLO) and event-centric (LOEO). In practice, sentiment columns were consistently out-ranked during pre-selection (top-K=20); showed near-zero or negative importance in permutation/SHAP tests; and were ultimately excluded from the final XGB.

Recommendations

Key actions for future model refinement and deployment

1. Broaden historical coverage
   - Secure paid data feeds to extend the back history and improve regime diversity
   - The upward trend in late-fold performance indicates additional history is likely to strengthen generalization across all three tickers.
2. Strengthen sentiment inputs
   - Upgrade to higher-quality, multi-source news feeds and filings.
   - Use a production OpenAI/API setup (rate limits, reliability, audit logs) and establish data-quality gates before integration.
3. Develop a pair-trading model
   - Train an XGBoost-based spread/mean-reversion model (e.g., GOOGL–AMZN) with cointegration checks and z-score entry/exit logic.
   - Evaluate with walk-forward tests and transaction-cost modeling.
4. Implement forward monitoring
   - Keep the model running in test mode, monitor simple metrics, trigger an alert when they worsen, and retrain based on clear cutoffs.
   - Continue feature/parameter sweeps and ablations to iteratively enhance performance.
5. Support ongoing model engineering
   - Refine feature families (event-aware gates/decays, rolling/EWM standardization), and re-optimize pruning (PI + stability + top-K).
   - Maintain reproducible notebooks, experiment logs, and versioned artifacts to support faster iteration and auditability.

Risks & Limitations

This analysis is subject to several constraints that may impact the accuracy and generalisability of findings.

- **First**, access to official earnings data is limited, which may necessitate reliance on secondary sources, potentially introducing bias or error.
- **Second**, if Natural Language Processing (NLP) methods are employed to extract insights from textual data (e.g., financial news or reports), the quality of results may be affected by model limitations in semantic understanding and contextual nuance.
- **Third**, the scope of historical data available for analysis may be restricted in terms of time span, granularity, or completeness. This can hinder the detection of long-term trends or cyclical patterns.
- **Additionally**, the computational demands of advanced analytical techniques, such as machine learning or large-scale simulations, may pose practical limitations. These processes often require significant processing power and can be costly, which may constrain scalability or real-time application.

Collectively, these limitations should be considered when interpreting results and considering deployment. Future work may benefit from improved data access, methodological refinement, and enhanced computational resources.

Appendix

Data Gathering Pipeline Flow

Files to be ran to prepare the Master Data file, in order:

1. Python - Master Data Pipeline Model Data Extraction.ipynb
2. (Optional, see note in file above) – SQL - Import_finnhub_data.sql
3. SQL - v02_stock_news_cleansing.sql
4. SQL - v03_stock_news_cleansing.sql
5. Python - OpenAI_sentiment_scoring.ipynb
6. (Optional, see note in file above) – SQL - Import_v02_and_v03_openai_sentiment.sql
7. SQL - Combining_v02_and_v03_plus_summaries.sql
8. SQL - stocks_master_preparation_file.sql

Methodological Notes (Boxes A.2–A.6)

BOX A.2: DE-MEANING AND STANDARDIZATION OF FEATURES

Most raw features in the dataset – including prices, volumes, and macroeconomic levels – introduce significant risks if incorporated directly into predictive modelling. They exhibit (a) ***strong drifts*** (long-run systematic upward or downward movements) – which can lead to *spurious trend learning* (i.e., the model interpreting shared trends as causal rather than focusing on deviations that contain predictive signals); (b) ***seasonality*** – which may be misinterpreted as a predictive driver of returns instead of highlighting when values are unusually high or low relative to their seasonally typical baseline; and (c) ***wide scale differences*** across variables and tickers, creating risks of *non-comparability* due to inconsistent units, and *volatility of coefficients* (if features are on very different scales, coefficients will be on wildly different magnitudes, leading to unstable estimation and unfair penalization under regularization).

To mitigate these issues, a two-step transformation was applied:

1. De-meaning (centring): most features were de-meaned using trailing rolling or exponentially weighted averages, shifted by one day to prevent look-ahead. This removes slow-moving drifts and regular seasonal components, forcing the model to focus on deviations from typical values rather than absolute levels.

For these features, de-meaning subtracts a trailing mean, centring the series around zero:

$$x_t^{\sim} = x_t - \mu_{t,w}, \quad \mu_{t,w} = \text{rolling/EWM mean over window } w \;\; \text{(shifted by one day)}$$

Slow-moving macroeconomic and earnings-level variables were normalized using global statistics (full-sample mean and standard deviation) without a rolling de-mean step.

2. Standardization (z-scoring): After centring, most features were standardized by dividing their de-meaned values by trailing volatility (standard deviation), computed on a rolling or exponentially weighted basis and shifted by one day.

$$z_t = \frac{x_t - \mu_{t,w}}{\sigma_{t,w}}, \quad \mu_{t,w} = \text{rolling/EWM mean}, \; \sigma_{t,w} = \text{rolling/EWM std}, \; \text{(all shifted by one day)}$$

This expresses each observation in units of standard deviations from its recent norm, which makes features directly comparable across time and across tickers, regardless of their original scale (price, ratios, macro levels, etc.). For families not de-meaned (macroeconomic levels and earnings levels), standardization was applied using global statistics. The result is that all features are placed on a common, volatility-adjusted scale (typically between −2 and +2 for most observations).

Detailed Methodological Documentation: targets (future returns), event flags (e.g., *event*, earnings_date_*), calendar/timing variables, and binary indicators were auto excluded from transformation. All remaining numeric variables were assigned to families based on their economic and statistical characteristics and for each family, tailored normalization methods were applied:

Rolling windows (≈60 trading days): Rolling normalization was applied to Price levels, Distance-to-trend/Momentum, and Market return features. A 60-day (~3 months) trailing window provides a stable local baseline while remaining responsive to medium-term changes, ensuring deviations are measured relative to a recent but robust benchmark.

Exponentially weighted moving averages (EWM): EWM normalization was applied to Volume/Turnover and Volatility/Beta features. Unlike rolling windows, EWM assigns greater weight to more recent observations, enabling faster adaptation to regime shifts in trading activity or volatility. For volume features, a half-life of 25 days was used, meaning the effective weight of an observation decreases by 50% every ~25 trading days. This choice ensures that the scale reflects the most recent trading patterns while still retaining a meaningful history. For volatility/beta features, a half-life of 40 days was chosen, providing a more gradual adaptation. This reflects the fact that volatility and beta tend to evolve more slowly than trading volumes, requiring a longer memory to establish a stable baseline.

Global statistics (full sample mean and standard deviation). Global normalization was applied to the slow-moving, low-frequency variables of Macroeconomic and Earnings level features, using a constant mean and standard deviation computed once over the full sample. This approach ensures stability for slow-moving, low-frequency series where short windows would generate noisy or artificial fluctuations. For Macroeconomic indicators, global statistics provide a consistent baseline for variables that change slowly and are typically released monthly or quarterly (e.g., CPI, GDP, unemployment). For Earnings levels, global normalization avoids overstating volatility between sparse quarterly updates while still putting different firms and earnings metrics on comparable scales.

Finally, to prevent look-ahead bias, all rolling and EWM statistics were computed on a trailing-only basis and shifted by one trading day. This ensures that the standardized value at time t uses information available only up to $t-1$, preserving the integrity of out-of-sample evaluation.

Contribution to modelling: by transforming features into rolling or exponentially weighted de-meaned and standardized variants, the dataset was rendered more stationary and better aligned for modelling. This process produced cleaner signals, enhanced robustness, and ultimately supported more consistent and reliable predictive performance.

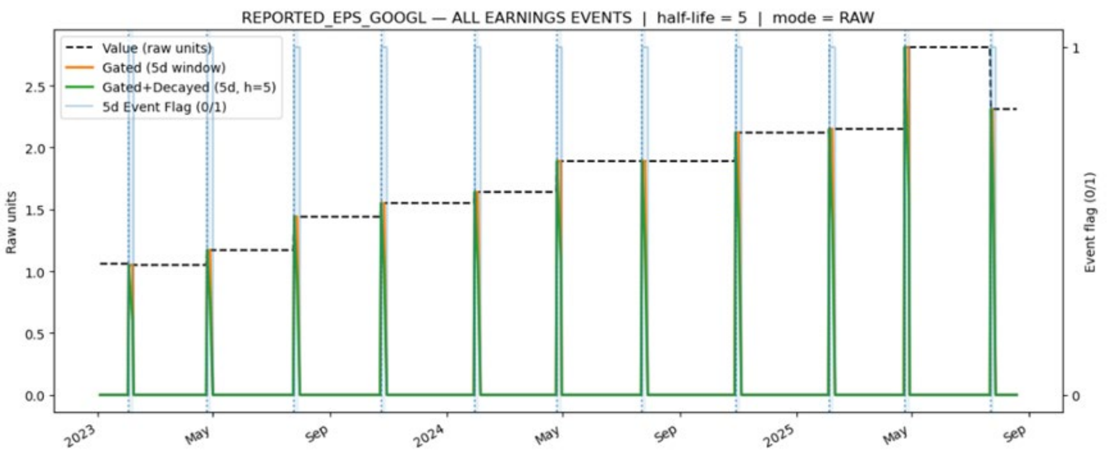
Box A.3: Event-Aware Feature Engineering

Macroeconomic variables (CPI, Trade, Unemployment, GDP, DSR, FOMC) and corporate earnings (EPS, revenue, surprises) are low-frequency events. If values are simply forward filled, the model sees the same number repeated for weeks or months. This creates two risks: (A) *Spurious trend learning* – the model might misuse a constant value as a proxy for time/trend; (B) *Dilution of impact* – the real market effect of a release is concentrated around the event, not uniformly spread until the next update. **To mitigate these risks and to properly align values with their true market impact, event-aware features were engineered**:

- **Event flags (0/1):** Mark days inside an event window (e.g. 1-day, 2-day, 5-day after release). *Example: earnings_date_5d_googl.*
- **Gated features:** Multiply the raw or standardized value by the event flag. *Example: reported_eps_googl__gate_earnings_date_5d_googl.*
- **Decayed features:** Multiply the raw value by a decay weight that halves every h days. *Example: reported_eps_googl__decay_h5.*
- **Gated+Decayed features:** Combine both. *Example: reported_eps_googl__gate_earnings_date_5d_googl_dec_h5.*

The process was applied to all macroeconomic features with half-lives tuned to horizon T+10 (7–20 days); and to earnings for each ticker with half-life = 5 days. Features were built for both raw and standardized (_gz) values. Raw columns were preserved (drop_raw=False) to avoid information loss and keep interpretability. Visualizations were produced to confirm the design and improve interpretability.

Example of Engineered Features — Raw EPS vs. Gated and Gated + Decayed Windows (GOOGL)



Contribution to Modelling: Event flags together with gated+decayed features eliminated misleading persistence, prevented the model from assuming that an EPS or CPI value should influence returns months later, and directed learning toward true event responses—capturing event-driven spikes in explanatory variables, reflecting the diminishing impact of releases, and avoiding overweighting of prolonged flat periods. The design allowed different horizons: 1d/2d gates – for short-term effects (T+2, T+5); and 5d gated+decayed – for medium-term effects and predictions (T+10). An EDA check confirmed that the new gated+decayed features show a significant correlation with T+10 returns. Finally, by transforming static low-frequency values into time-aware, event-sensitive features, modelling produced cleaner signals, reduced spurious trend dependence, and enhanced short- to mid-term predictive power.

BOX A.4: SELECTION OF FEATURES AND PRUNING METHODS

Modelling features in this project are intentionally rich: target-specific signals, peer information, macro and earnings releases, and their event-aware transforms (gates/decays), typically present in both raw and standardized variants. Left unfiltered, this breadth risks variance inflation – weak, redundant, or degenerate columns that add noise and might hinder performance rather than add predictive power. To compress and optimize the feature space, a robust, multi-level pruning process was applied using complementary methods:

1. Deterministic prefiltering: within each path**: (a)** standardized twins were preferred (if a raw level and a standardized/de-meaned version coexist, the raw is dropped and the standardized variant is kept); while **(b)** a lightweight correlation screen prunes features whose association with the target is negligible: columns with absolute Pearson (and, when invoked, absolute Spearman) below **0.10** or undefined due to missing overlap are discarded.. In practice, this step *eliminates families of near-random event gates and miscellaneous low-signal artifacts* while leaving intact standardized, economically grounded signals.

2. Leakage-safe model-based pruning: two complementary selectors follow pre-filtering:

(A) Permutation Importance (PI): *w*ithin each outer fold, the training window is split chronologically using `*val_frac = 0.25*` into (i) a *sub-train slice*—the earlier 75% used to fit a lightweight RidgeCV scoring model—and (ii) an *inner validation slice*—the later 25% used only for evaluation and entirely preceding the fold's test window. On the inner validation slice, each feature is shuffled in isolation; predictions are recomputed, and the resulting drop-in hit rate is recorded. The shuffle is repeated several times per feature and averaged to yield a per-fold PI score. PI is expressed in percentage points of hit (e.g., PI = 0.01 implies a 1pp average reduction in validation hit when the feature is destroyed). PI scores are then averaged across outer folds, and features with mean Δhit at or below the chosen cutoff (e.g., ≤ 1pp) are removed. Unlike correlation, which reflects pairwise association, PI quantifies incremental contribution given the rest of the feature set; a column can appear correlated yet contribute zero—or negative—Δhit once competing predictors are present. This method systematically removes features that harm validation performance, yielding a cleaner, more parsimonious set before any model is evaluated on the fold's test window.

(B) Stability selection with Ridge. For each outer fold, a RidgeCV model is fitted on the fold's training window and features are ranked by the absolute value of their coefficients (larger |coef| ⇒ stronger contribution within that fold). The top 40 features from each fold are recorded. After processing all folds, a feature is kept only if it appears in at least 60% of the folds. This rule rewards predictors that recur across time and filter out one-off, regime-specific spikes. Because the ranking is computed using only the training window in each fold, the selection remains leakage safe.

The two filters are combined conservatively: first PI-neutral/harmful columns are dropped, then intersect with the stability keepers. If this intersection becomes too thin to learn from, a safety floor restores a "drop-harmful-only" set so the path remains trainable.

3. Top K feature selection for XGB (hybrid step): after PI and stability selection, the remaining features form a compact, time-robust pool. The pipeline then fixes **K per path** so that XGB trains only on the strongest slice of this pool. This stage uses the cached per-fold Ridge rankings *(one ordered list of features per fold, produced on that fold's training window by RidgeCV and saved for reuse—features are sorted by absolute coefficient, highest first)*, which order the surviving features. Next, a small grid of K values (e.g., 20, 30, 40, 60) is tried (using the Fast K-sweep). For each fold and each K, only the top-K in that fold's cached order are passed to a lightweight, early stopping XGB that is tuned on the time-ordered inner validation slice of the training window and then scored on the fold's test window. Metrics are recorded in hit rate (primary) and Spearman rank correlation (secondary). Because rankings are cached, the sweep is computationally cheap and remains leakage safe. The K that yields the highest average hit rate across folds is chosen; ties are broken by higher average Spearman, and if still tied, by the smaller K for parsimony.

Contribution to modelling. The selection-and-pruning pipeline *reduces variance without amputating core signal*. Standardization ensures comparability; the correlation screen trims obvious dead weight; PI removes features that fail to move validation hit; stability selection enforces time-robustness; and the hybrid Ridge/XGB interface—by training XGB only on a fold-appropriate feature set—focuses non-linear capacity on a curated subset. All filtering steps live inside a walk-forward protocol with an embargo equal to the prediction horizon, ensuring realistic out-of-sample timing (no training label depends on outcomes from the test window). *The result is a smaller, cleaner, and more stable feature set for each path*, accompanied by supporting documentation: PI tables, lists of removed "harmful/neutral" columns, stability counts, and the final feature inventories used for training.

Referenced Functions (by name): *build_model_frame, prefer_standardized, drop_unrelated_features, audit_corr_nan, audit_lowcorr, apply_model_based_filters, perm_importance_by_fold_val, ridge_stability_select, cache_ridge_rankings, sweep_K_with_cached_rankings, pick_best_K, run_path*

BOX A.5: WALK-FORWARD CROSS-VALIDATION

The walk-forward evaluation framework provides a time-robust assessment of model performance that a single random end split cannot. While a conventional 80/20 (or 75/25) split is leakage-free, it concentrates evaluation in *one recent window* it does not establish whether the model generalizes across time. To address this, a walk-forward cross-validation regime was adopted. The walk-forward protocol executes *a sequence of chronologically ordered train–test experiments*, evaluating the model across several market states rather than a single recent period. While this approach may lower headline hit rate, it reduces false confidence and error risk, yielding *more reliable performance estimates*. It also reveals regime sensitivity – folds that span macro or event shocks (e.g., tariffs, wars, Federal Reserve actions) often show dips in hit rate – information that a single split would average away.
In this project, folds use approximately one trading year for training, one quarter for testing, and advance by roughly one trading month between folds. An embargo equal to the prediction horizon removes rows whose forward returns would overlap the test labels, eliminating look-ahead via overlapping targets. The selection–pruning blueprint governs each fold, including leakage-safe, model-based pruning in training (see Box A.4)

Evaluation is reported per fold and in aggregate
(A) Directional accuracy (hit rate) measures the share of test-set days where the sign of the prediction matches the realized sign at the chosen horizon.
(B) Spearman's ρ captures the monotonic alignment of scores with outcomes.
(C) The conditional long-only return (`strat_ret`) reports the average realized return on the subset of test days for which the model's prediction is positive.
For each metric the cross-fold mean, and its variability are provided, while the most recent fold is highlighted as the operational proxy because it is trained on the largest history and assessed in the latest market regime. *In equity forecasting* at ten-day horizons, mean hit rates in the low-to-mid fifties

are typically commercially relevant when stable, Spearman values above roughly 0.10 are notable, and conditional long-only returns that materially exceed unconditional averages indicate useful selection before costs.

Overall, the walk-forward regime – combined with path-aware feature scoping, deterministic prefilters, leakage-safe model-based pruning, and a compact hybrid XGB model—produces an evaluation that is both realistic and defensible. It measures *generalization across changing regimes*, enforces *timing integrity* through embargo and warm-ups, and offers *two complementary perspectives on performance*: while the last fold uses the most history and tests on the most recent regime – providing a benchmark for live short-term deployment – the cross-fold mean hit, averaged across many past regimes, supplies a long-run baseline when the future state is unknown.

BOX A.6: LEAVE-ONE-EARNINGS-OUT (LOEO)

LOEO provides earnings-respectful validation by holding out a single earnings window and evaluating whether a model trained on all remaining days (*including past earnings windows*) can accurately predict targets in that unseen event window. Earnings anchors (the date where earning _date == 0) were identified using the "days-since-earnings" counters on the wide event-engineered dataset for the ticker (e.g., wide_google_aug_tied). Around each anchor (A) a test window of $\pm W$ business days is defined, centred on the event day (when *W=2 yields five test days (A−2,…,A+2); W=5 yields eleven; W=10 yields twenty-one*). To prevent look-ahead from the prediction horizon, an embargo of H trading days is applied on both sides of that test window. In practical terms, the training set for that fold consists of every date outside the window and the embargo [$A-W-H$, $A+W+H$]; the test set is exactly the $\pm W$ block.

Similar to the walk-forward CV regime, performance is computed per earnings fold and then aggregated, using the same three evaluation metrics: hit rate, Spearman's ρ, and conditional strategy return. As before, reporting both the cross-fold mean, and the last (most recent) fold distinguishes long-run robustness from current-regime performance.

LOEO complements walk-forward CV. Combined, they couple robustness across market regimes with evidence of generalization to the next earnings event, providing a complete and realistic read on deployable performance.

Risks & Limitations:

1. *Early anchors may be dropped at alignment*: when initial earnings dates fall within the warm-up window used for rolling standardization and safeguards, LOEO excludes those folds by design. This reduces the number of evaluable events. Mitigations include extending the historical span so ~8–12 earnings events remain after alignment; in this project, at least nine folds were evaluated per company.
2. *Tight LOEO windows yield small test samples*: with few observations per fold, metrics such as hit rate and Spearman's ρ can fluctuate. In this setting, RidgeCV often remains competitive (lower variance helps when data per fold are scarce), whereas XGB can exhibit fold-to-fold volatility in hit/ρ. Mitigations include using modest (but not tiny) windows (e.g., ±5 or ±10), reporting confidence intervals across folds, and complementing LOEO with walk-forward CV to show performance outside the earnings window. Around earnings, the hybrid XGB approach particularly benefits from Ridge-based top K pre-selection, which serves as a valuable safeguard.

Glossary

A

Adjusted Close Price: The stock's closing price adjusted for dividends, splits, and other corporate actions to reflect true historical value.

Amortisation: The gradual reduction of a debt or the cost of an intangible asset over time.

Asset: Any resource owned by a company that has economic value and can be used to generate future benefits.

B

Balance Sheet: A financial statement that shows a company's assets, liabilities, and shareholders' equity at a specific point in time.

Book Value: The net value of a company's assets minus its liabilities.

Balance Sheet: A financial statement that shows a company's assets, liabilities, and shareholders' equity at a specific point in time.

Beta 60-Day (beta_sp_60d**)**: Measures a stock's sensitivity to market movements (SPY) over a 60-day rolling window. Quantifies sensitivity to the market—how much the stock typically moves when the index moves—estimated over \~60 days. It is central to hedging (using offsetting positions—e.g., futures - standardized contracts to buy/sell later at a set price, options - rights—not obligations—to buy/sell at a set price by a date, or shorts - selling borrowed shares to profit if the price falls —to reduce or neutralize losses from adverse price moves); and portfolio construction (choosing assets and weights - the % of total capital allocated to each holding - to reach a target return/risk profile, subject to constraints such as diversification [spreading exposure to avoid concentration risk], sector limits [caps on how much can be invested in any one industry], and turnover [how frequently positions are changed, which drives trading costs]). Values > 1.2 behave aggressively with the market; \~1.0 is market-like; < 0.8 is defensive (tends to move less than the market). Beta can drift with regime changes (rates, policy, business mix) and tends to shift across quarters rather than single event days; earnings seasons can alter beta as new information changes how a name comoves with the index.

Book Value: The net value of a company's assets minus its liabilities.

Bollinger Bands: A technical analysis tool with three lines: a middle band (20-day SMA), an upper band (+2 standard deviations), and a lower band (−2 standard deviations).

Bollinger Lower (bollinger_lower**)**: Lower boundary of Bollinger Bands.

Bollinger Position (bollinger_position**)**: Normalised position of the current price within the Bollinger Bands. bollinger_position (normalized location within 20-day Bollinger bands) . A scale-free reading of where today's close sits inside its recent 20-day price envelope. Using the 20-day average with bands at ±2 standard deviations, the metric reports how far today's close has progressed toward the upper band across the full width from the lower band to the upper band: it equals (today's close − lower band) divided by (upper band − lower band), yielding a 0–1 "closeness-to-upper-band" scale. Values near 1.0 indicate price hugging the upper band (extended strength), near 0.0 indicate closeness to the lower band (pressure/weakness), and ~0.5 is mid-range/neutral. Bands widen when volatility rises and narrow during quiet periods; persistent readings near the extremes across several sessions point to sustained trend rather than a one-off spike. Around earnings or major news, a durable shift from low to high positioning (or the reverse) often accompanies a regime change in the name's short-term behaviour.

Bollinger Upper (bollinger_upper**)**: Upper boundary of Bollinger Bands.

C

Cash Flow: The movement of money into and out of a business, categorised into operating, investing, and financing activities.

Capital Expenditure (CapEx): Funds used by a company to acquire or upgrade physical assets such as property or equipment.

Company Sentiment Toward AI: can be derived from earnings call transcripts — verbatim records of quarterly analyst calls. These transcripts reveal how companies communicate about AI, capturing tone, strategic focus, and sentiment shifts. Unlike formal filings, they reflect real-time executive positioning, making them well-suited for NLP-based sentiment analysis.

D

Daily Returns (ret_1d**)**: Percentage change in price from one day to the next. Measures how far price sits below its highest close in the last 60 days, summarizing the depth of a recent slump. Near 0 indicates price is at/near highs; −10% to −20% marks a notable correction; < −30% reflects a deep slump. Large drawdowns often follow negative earnings/guidance cycles, sector de-rating, or macro headwinds; ongoing deterioration together with elevated downside volatility generally signals a risk regime that warrants closer monitoring.

Depreciation: Allocation of the cost of a tangible asset over its useful life.

Dividend: A portion of a company's earnings distributed to shareholders.

E

Earnings Per Share (EPS) Net income divided by the number of outstanding shares; indicates profitability per share.

Earning Surprise (%): Difference between actual and expected financial results.

Equity: Ownership interest in a company.

Estimated EPS: Analyst forecast of earnings per share.

F

Form 10-K: An annual report filed with the SEC that includes audited financial statements and detailed company information.

Form 10-Q: A quarterly report filed with the SEC that includes unaudited financial statements and management analysis.

G

GAAP (Generally Accepted Accounting Principles): Standard framework of guidelines for financial accounting used in the U.S.

gap_ret (overnight gap return): The after-hours/overnight move from yesterday's close to today's open. Gaps isolate information arriving outside regular hours (earnings, M&A, guidance, macro headlines). Large positive gaps (>+2–5% in large caps) often reflect surprises to the upside; large negative gaps reflect disappointments. Some names continue in the gap direction (trend days), others mean-revert (price drifts back toward the prior close)—this varies by sector and catalyst. Repeated gaps in one direction within a quarter can indicate a changing fundamental narrative or positioning/flow dynamics.

Gross Profit: Revenue minus the cost of goods sold (COGS).

I

Idiosyncratic Volatility 20-Day (idio_vol_20d**)**: Volatility not explained by market movements. The volatility that remains after removing market moves- highlighting company-specific risk and news flow. Higher readings (≈ 1–4% per day on residuals, sometimes higher in event names) indicate that the stock is jumpy on its own; lower readings indicate motion is mostly market-driven. Elevation is commonly linked to earnings cycles, product/clinical readouts, management changes, M\&A headlines, or litigation; persistent idiosyncratic volatility often signals an ongoing catalyst path.

Income Statement: Shows revenues and expenses over time.

Interest Expense: Cost of borrowed funds.

Interest Expense The cost incurred by an entity for borrowed funds.

L

Liability: Legal financial obligations. Liquidity: Ability to meet short-term obligations.

Liquidity / Participation: how unusual current trading activity is relative to the stock's own history, as a proxy for attention and participation intensity. Example: volume z-score (60-day).

Lower Band: Lower boundary of Bollinger Bands.

M

macd (12/26-day EMA spread) : The difference between a fast and a slow-moving average of price. Computed as the 12-day EMA minus the 26-day EMA, where each EMA is updated recursively as $EMA_t = \alpha \cdot close_t + (1-\alpha) \cdot EMA_{t-1}$ with $\alpha = 2 / (span+1)$ (so $\alpha_{12} = 2/13$, $\alpha_{26} = 2/27$). positive and rising values indicate the fast trend is above—and pulling away from—the slow trend (building upside momentum), while negative and falling values indicate the opposite (building downside momentum). Crossings around zero often mark potential shifts in the dominant swing, and multi-day persistence carries more weight than a single print. Earnings and major news can flip MACD abruptly; evidence of follow-through in subsequent sessions typically distinguishes trend change from headline whipsaw.

Market Capitalisation: Total value of outstanding shares.

Market Cap (B): Market cap expressed in billions.

Momentum (Position & Trend): where price sits now within a short-term band and how much directional progress has accumulated over recent windows. Examples: RSI, Bollinger position, 20/63-day momentum, 20-day Sharpe, MACD. Momentum is one of the most studied effects in finance. These features describe both where price sits inside its short-term band (position: e.g., RSI, Bollinger position) and how much directional progress has accumulated over recent windows (trend: e.g., 20/63-day momentum, 20-day Sharpe, MACD). Together they help distinguish steady compounders with persistent, orderly advances (Apple-like) from bursty, momentum-driven names that surge or stall in waves (Nvidia-like), while also indicating when price is stretched or extended relative to its recent range.

Momentum 20-Day (mom_20d**)**: Price change over 20 trading days. Sum of the last \~20 trading-day returns. Captures the short-term trend—specifically, whether there has been a sustained move (many days in the past month with returns consistently above or below zero, not a single spike). It's widely used for screening (fast filter for recent strength/weakness), timing (enter/exit when the sign turns and holds), and regime tagging (label a name as uptrend / sideways / downtrend for downstream models). Higher/positive (often roughly +0.05 to +0.25 ≈ +5–25%) means the stock has been rising over the month; lower/negative (≈ −5–25%) means it has been falling. Extremes often cluster around earnings surprises, guidance changes, and major news. As a rule of thumb, a one-day flip is not considered actionable; assessment typically waits for 5–10 days of persistence. Readings at or below −10% that continue to worsen—especially alongside rising downside volatility or deepening drawdown—are commonly regarded as a red-flag condition.

Momentum 63-Day (mom_63d**)**: Price change over 63 trading days. Sum of the last ~63 trading-day returns (about three months). Captures the medium-term trend—that is, whether there has been a sustained move (many weeks in which returns are predominantly above or below zero, not a one-week pop). It's popular in professional screening and portfolio construction because three-month momentum is more stable than very short windows, lines up with the quarterly earnings cadence, and helps tag regimes (multi-week uptrend vs. downtrend) for downstream models or risk overlays. Higher/positive values (typically ~+10% to +40%) indicate a multi-week advance; lower/negative values (roughly −10% to −40%) indicate a multi-week decline. Large magnitudes often reflect repeated earnings beats/misses, upgrades/downgrades that accumulate over weeks, product cycle narratives, or macro tailwinds/headwinds affecting the sector. Rather than acting on a single reversal day, decisions are typically deferred until the sign of mom_63d has held for ~10–15 trading days. Readings at or below −15% that continue to deteriorate—especially alongside rising downside volatility or deepening drawdown—are commonly treated as a red-flag condition.

MD&A (Management's Discussion and Analysis) A section in SEC filings where management discusses financial results, trends, and outlook.

N

Net Income: The company's total earnings, calculated as revenue minus all expenses.

Net Margin: Net income divided by revenue; shows overall profitability.

O

Operating Income: Earnings before interest and taxes (EBIT); reflects profit from core operations.

Operating Margin: Operating income divided by revenue; measures operational efficiency.

Operating Margin (%): The percentage of revenue left after paying for core operating costs (before interest and taxes). Formula: Operating Income ÷ Revenue × 100. Preferred over profit margin as it excludes tax and financing noise.

Q

Quarter Ending: Final date of fiscal quarter.

R

range_pct (intraday range as % of close): A same-day turbulence metric capturing how far price moved intraday. Measures the intraday distance between the day's high and low, divided by the closing price. Higher readings (large fraction of price, e.g., 3–10% in big names; more in smaller caps) indicate busy, headline-driven or breakout days ("breakout" = a strong move beyond recent highs/lows); lower readings (<1–2% in large caps) suggest a contained session. Because it is normalized by price, it compares across tickers and through time. Surges commonly cluster around earnings releases, guidance changes, regulatory/macro prints, and news-led squeezes; repeated high values over several sessions can indicate ongoing repricing rather than noise.

Realised Volatility 20-Day (rv_20d**)**: Annualised standard deviation of returns.

Reported EPS: Actual earnings per share reported.

Return: Percentage change in asset price.

Return on Equity (ROE): Net income ÷ shareholders' equity.

Return Volatility: Fluctuation in daily returns.

Return Volatility 5-Day (return_volatility_5day**)**: 5-day standard deviation of returns.

ret_1d (1-day return): The single-day price change, a micro-momentum signal. Large positive or negative values typically mark event days (earnings, guidance, macro headlines). It is the atomic input for many rolling features; on its own it is noisy, but in aggregation it helps characterize trend bursts and shocks.

ret_5d (5-day return): The cumulative move over ~one trading week, capturing very short-term trend. Elevated positive values indicate recent follow-through; large negatives indicate recent

Revenue (Million): Total income before expenses expressed in millions.

Revenue Growth (YoY): Year-over-year revenue increase.

Relative Strength Index (RSI, 14days, 0-100): a widely used momentum oscillator in technical analysis that helps traders assess whether a stock is overbought or oversold. A short-horizon momentum oscillator that summarises whether up days or down days have dominated over ~14 sessions. It maps Relative Strength (smoothed 14-day average gain ÷ smoothed 14-day average loss) to a 0–100 index via RSI = 100 − 100/(1 + RS). RSI≳70 reflects an unusually strong run ("overbought" = stretched to the upside, not a guaranteed reversal), RSI≲30 reflects an unusually weak run ("oversold"). For trend context, sustained >60 is often associated with healthy uptrends and sustained <40 with downtrends. Post-earnings or headline weeks can push RSI to extremes; the key signal is whether it stays elevated/depressed for several days (trend) or quickly snaps back (noise).

rv_5d (5-day realized volatility) : Standard deviation of daily returns over \~1 week. Provides a quick read on very recent turbulence. It is useful for short-horizon risk sizing and for flagging event weeks. Higher readings (≈ ≥ 3% per day) indicate a jumpy week; lower readings (≈ 1–2% per day) indicate a calm week. Spikes commonly appear around earnings releases, guidance updates, regulatory headlines, or macro prints; interpretation typically emphasizes persistence over several sessions rather than reacting to an isolated day.

rv_20d (20-day realized volatility): Month-level volatility. A widely used baseline risk gauge (a standard, go-to measure of typical price variability over a recent window) for position sizing (choosing how big the investment should be—how many shares or how many dollars—to keep potential losses within agreed limits) and Value-at-Risk (a statistical estimate of the maximum expected loss over a chosen horizon at a specified confidence level, e.g., 95% one-day VaR). Higher levels (≈ 2–6% per day) signal larger routine swings; lower levels signal steadier trading. Elevation often coincides with earnings seasons, prolonged news cycles, sector rotations, or macro uncertainty. In practice, trend context matters: the same volatility is viewed differently in a steady up-trend versus a disorderly tape.

S

semivol_20d (20-day downside volatility): Volatility of loss days only, focusing on downside risk rather than overall choppiness. Higher values indicate that down moves are large when they occur; when semivol_20d sits close to rv_20d, most variability is coming from losses. Elevation is frequently associated with negative earnings surprises, guidance cuts, downgrades, or adverse regulatory news. Persistent elevation over several weeks is generally treated as a sign of deteriorating risk quality.

Shareholders' Equity The residual interest in assets after deducting liabilities; represents ownership value.

sharpe_20d (20-day "quality of trend") : Average daily return over the last ~20 trading days divided by the daily return volatility over the same window—a short-horizon Sharpe ratio (i.e., return per unit of volatility over ~1 month; the risk-free adjustment is typically negligible at daily horizons and is omitted). It captures the quality of the recent trend by asking how much return you're getting per unit of noise: smooth, orderly advances score higher than jittery ones with the same net gain. This metric is common in screening and portfolio sizing because it helps prioritize exposures that are both productive and stable. Higher/positive values (often ~0.2–1.0 in calm uptrends, occasionally >1.5 during exceptional runs) indicate a steady, well-behaved advance; near zero or negative means the period is choppy or losing, even if some days were strong. Around earnings/news weeks, volatility typically jumps; that can pull sharpe_20d down even when price ends higher. Rather than reacting to a single print, interpretation should wait for sharpe_20d to recover and remain positive for several days.

SEC (Securities and Exchange Commission): U.S. government agency responsible for enforcing federal securities laws and regulating the securities industry.

Stock Returns (high/low) Measured over a fixed pre-event window (e.g., 7 days prior to earnings) to capture market momentum while avoiding post-event data leakage; trailing return windows of 3, 7, or 14 days can be tested.

SEC: U.S. securities regulator.

Semivariance 20-Day (semivol_20d**)**: Downside risk measure.

Sharpe Ratio 20-Day (sharp_20d**)**: Risk-adjusted return metric.

Shareholders' Equity: Residual interest in assets.

SPY: ETF tracking the S&P 500.

Stock Returns (high/low): Pre-event return windows.

Surprise Percent: Difference between Reported and Estimated EPS.

T

Tickers: Stock symbols (e.g., AAPL, NVDA).

U

Upper Band: Upper boundary of Bollinger Bands.

V

Volatility/Risk: Degree of price variation. How large and asymmetric price moves typically are, how tightly the stock co-moves with the market versus its own idiosyncratic behaviour, and how far it sits from recent peaks. Examples: 5/20-day realised volatility, downside (semi)volatility, drawdown (60-day), beta to S&P (60-day), idiosyncratic volatility, plus single-day movement descriptors such as intraday range % and overnight gap return.

Volume: Number of shares traded. volume (raw shares traded). A scale-dependent measure of trading activity and capacity to transact. Raw volume is useful within a ticker through time (e.g., rising or falling participation); for cross-ticker comparison, a normalized measure such as vol_z_60d is preferred. when mixing volume with other features, a per-ticker z-score or a log transform (np.log1p(volume)) avoids size effects from dominating.

Volume Z-Score 60-Day (vol_z_60d**)**: Standardised volume comparison. A z-score of today's trading volume relative to the past ~60 days: it tells how many standard deviations today is above or below the recent average (z≈0 = typical day; z=+2 means today's volume is ~2 SDs above normal; z=−2 is ~2 SDs below). Unusual volume often indicates a change in market participation (the number and mix of active buyers/sellers) and liquidity (the capacity to trade size quickly with minimal price change). High positive values (≈ +2 and above) flag attention spikes—often around earnings, guidance, product/news drops, index rebalances, or options events; deep negatives (≈ −2 or below) indicate unusually quiet sessions. Persistent elevation over several days can signal sustained interest/flow, whereas one-day spikes often fade. In practice, vol_z_60d is used to screen for attention spikes (systematically surface names with abnormal interest); flag event windows (days around earnings, guidance, regulatory or macro releases, index rebalances, or options expiries when trading conditions differ from normal); validate price moves (advances/declines accompanied by high volume are more likely to reflect broad conviction than moves on light volume—useful for confirming "breakouts" or "selloffs"); prioritize monitoring when managing a large universe (dozens to hundreds of tickers); and set liquidity-aware execution plans. Note: holiday weeks and structural changes (e.g., splits) can shift baselines and should be considered when interpreting z.

External Sources

- NVIDIA Corporation - Financial Reports, viewed August 2025, <https://investor.nvidia.com/financial-info/financial-reports>.
- Investor Relations - Apple, viewed August 2025, <https://investor.apple.com/investor-relations/>.
- Investor Updates - Alphabet Investor Relations, viewed August 2025, <https://abc.xyz/investor/>.
- Investor Relations - Analog Devices, viewed August 2025, < https://investor.analog.com/>.
- Investor Relations | Autodesk, Inc. , viewed August 2025, < https://investors.autodesk.com/ >.
- Advanced Micro Devices, Inc. (AMD) , viewed August 2025, < https://ir.amd.com/ >.
- Amazon.com, Inc. - Overview, viewed August 2025, < https://ir.aboutamazon.com/overview/ >.
- Investor Center | Broadcom Inc. , viewed August 2025, < https://investors.broadcom.com/ >.
- CDW LLC - Events & Presentations, viewed August 2025, < https://investor.cdw.com/events-and-presentations/ >.
- Investor Relations | Cognizant, viewed August 2025, < https://investors.cognizant.com/home/ >.
- Lam Research Investor Relations - Investors Overview, viewed August 2025, <https://investor.lamresearch.com/>.
- Investor Relations | Marvell Technology, Inc., viewed August 2025, < https://investor.marvell.com/>.
- Microsoft Investor Relations - Company Overview and Strategy, viewed August 2025, < https://www.microsoft.com/en-us/investor/company-overview>.
- Investor Relations | Micron Technology, viewed August 2025, < https://investors.micron.com/>.
- Investor relations | TI.com, viewed August 2025, < https://investor.ti.com/>.
- SEC.gov | Search Filings, viewed August 2025, <https://www.sec.gov/search-filings>.
- Google Finance – Stock-market prices, real-time quotes and business news, viewed August 2025, <https://www.google.com/finance/>.
- Yahoo Finance - Stock Market Live, Quotes, Business & Finance News, viewed August 2025, < https://finance.yahoo.com/>.
- St. Louis Fed Web Services: FRED® API, viewed August 2025, < https://fred.stlouisfed.org/docs/api/fred/>.
- The Fed - Meeting calendars and information, viewed August 2025, < https://www.federalreserve.gov/monetarypolicy/fomccalendars.htm>.
- Finnhub Stock APIs - Real-time stock prices, Company fundamentals, Estimates, and Alternative data., viewed August 2025, < https://finnhub.io/>.
- developer.nytimes.com/apis, viewed August 2025, < https://developer.nytimes.com/apis>.