THE LONDON SCHOOL OF ECONOMICS AND POLITICAL SCIENCE

FOURTH REV

VANTAGE DATA

INSIGHT ALCHEMISTS
[EP - Team 1]

DA401- Employer Project (VP Analytics)

Decoding Stock Returns: The Role of Earnings, Macro factors, Financial Metrics and Sentiment

Technical Report

24/02/2025

Insight Alchemists [Team 1]
Hemalatha Nagaraj

Manasa Chilakapati

Sagar Varma

Sujith Kumaar K C

Yuliya Pauzunova

Contents

1. Problem Statement:

VP Analytics seeks insights to optimizing their trading strategies around earnings announcements for major tech companies (NVIDIA, Apple, and Google)

Stakeholders: VP Analytics (John) & Hedge Fund Partners, Fourth Rev (Andre, Danni, Abi), LSE Partners, Teammates.

2. Business Objective Analysis:

Through a structured approach analysis of the business objective as shown here in the Ishikawa diagram – the business objective was defined.

Develop a comprehensive data-driven that decodes the relationship between stock price movements and earnings announcements, enabling hedge funds to optimize their trading strategies through predictive insights. This framework aims to integrate pre-earnings price patterns, financial metrics (earnings surprise, revenue, margins), macroeconomic indicators (inflation, GDP, interest rates), and news sentiment to generate actionable trading signals.

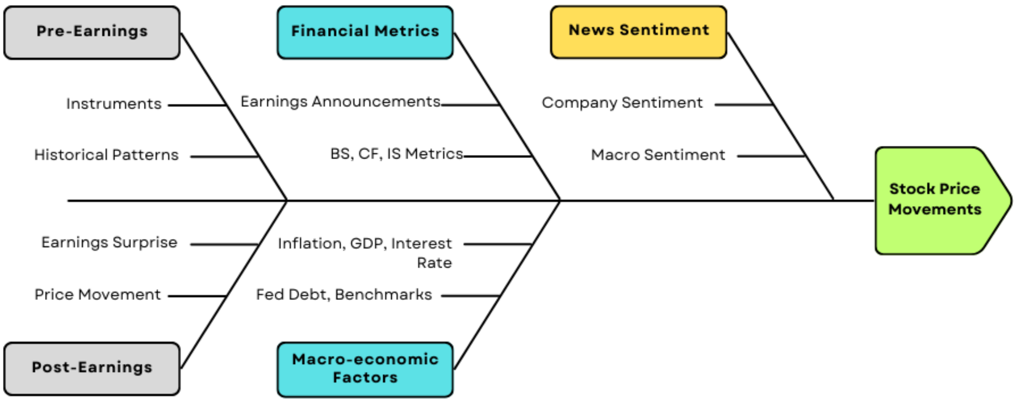


Fig 1: Ishikawa Diagram - Objective Analysis

3. Analysis

#	Python Notebook	Task	Analysis Explanation
1	**Data_Preparation_Team1_VP .ipynb**	Ingesting Data from Multiple API sources, Feature Engineering and providing CSVs utilized by other notebooks.	3.1
2	**Sentiment_analysis_Team1_VP .ipynb**	Sentiment Analysis on provided Articles and Stock prediction modelling based on Sentiment	3.2
3	**EDA_Macros_Earnings_Team1_VP .ipynb**	Exploratory Analysis on Earnings Data, Macroeconomic parameters against Stock metrics.	3.3 and 3.4
4	**Trading_strategy_ predictive_modelling_Team1_VP.ipynb**	Final Trading Strategy using Random Forest model, utilizing insights from EDA and Sentiment	3.5

3.1 Data Collection and Preprocessing

Python Notebook to refer: **Data_Preparation_Team1_VP.ipynb**

3.1.1 Data Source Identification

Key Rationale for Data Source Identification are provided below along with the finalized data sources-

- Time Period Coverage: Initially set for 2009-2024, with flexibility to adjust between 2000-2024 based on analysis requirements
- Source Prioritization: Preference given to:
    - Direct data sources (FRED) over aggregators
    - Pre-calculated metrics over raw data requiring complex processing
    - Comprehensive historical coverage over partial data availability

Data Category	Initial Source	Final Source	Key Decision Rationale
Stock price data (OHLCV)	YFinance	YFinance	• Simple extraction to Python dataframe • 20+ years of historical data available • Reliable, simple and consistent data quality
Earnings & Financial Report Metrics	YFinance AlphaVantage	Financial Modelling Prep (Pro)	• YFinance limited to 2022-2024 • AlphaVantage had raw data and required calculations of financial metrics • FMP Pro version provided pre-calculated metrics – reducing complexity in data processing
Macro-economic Indicators	AlphaVantage FMP Pro	FRED API	• Direct access to source data • Comprehensive coverage including Federal Debt • More granular control over data extraction
Market Sentiment	N/A	Vantage Data	• Client-provided dataset (2019-2024) of NY Times articles.

3.1.2 Data Wrangling and Manipulation

Data sourcing and preprocessing were centralized in a single notebook that generated standardized CSV outputs. This pipeline integrated multiple API sources (yFinance, FMP, FRED) and handled all cleaning and feature engineering, ensuring consistency across all downstream analyses.

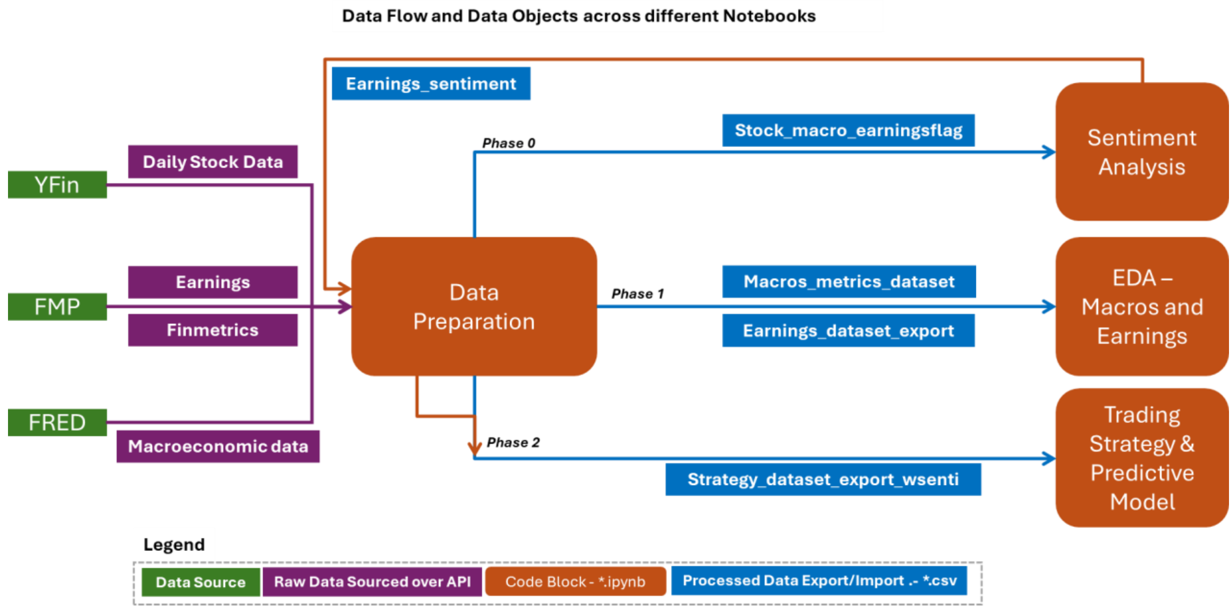


Fig 2 : Data Sourcing and Pre-processing pipeline flowchart

Data Processing and Feature Engineering

Quality Assurance & Standardization

- Missing data was forward-filled and outliers removed to maintain data integrity
- Earnings announcement dates were standardized to AMC convention for consistent analysis

Feature Engineering (90 derived features calculated – Refer Appendix 2)

- Stock metrics (returns, BOP, window returns) were calculated to capture price dynamics
- Financial indicators (earnings/revenue surprises) were derived to measure company performance
- Macro factors (GDP, CPI, Treasury yields) were integrated to reflect market conditions
- Temporal features (TTM, QoQ, YoY) were computed to account for seasonality patterns

Processing Decisions

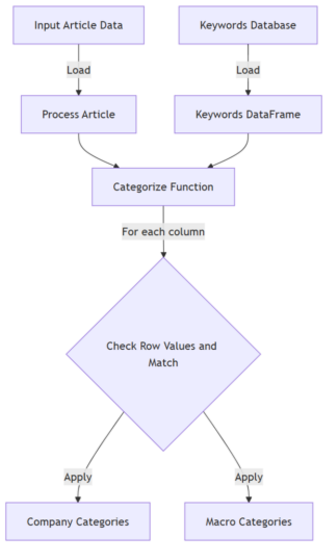
- Low-frequency observations were removed to ensure reliable signal quality
- Negative earnings estimates were set to NaN to avoid misleading surprise percentages
- Multi-source data was merged to enable integrated cross-dimensional analysis

3.2 Sentiment Analysis

Python Notebook to refer: **Sentiment_analysis_Team1_VP.ipynb**

3.2.1 Data Wrangling and Preprocessing

- Initial Data Load: The dataset was loaded from the provided file containing articles with various attributes after 51 articles were removed and character encoding issues resolved.
- The publication date was split into separate day and time columns, and additional month, year columns were created.
- The 'headline' column was parsed to extract 'main' and 'print_headline' values.
- **Keywords** were extracted and a **rule-based classification logic** applied to categorize articles into Apple/Google/NVIDIA and Macro news (Tech-AI/Tech-Macro/Market-Macro).
- Ensured **scalability** of the classification logic by maintaining an external sentiment_keywords.csv for easy updates. This workflow allowed for categorizing articles into relevant sector and organizations.



Keyword	Company_value	Macro_value
Apple Inc	Apple	
Apple TV	Apple	
Google Inc	Google	
Alphabet Inc	Google	
NVIDIA Corporation	Nvidia	
Artificial Intelligence		Tech-AI
ChatGPT		Tech-AI
Cloud Computing		Tech-Macro
Computer Chips		Tech-Macro
Microsoft Corp		Tech-Macro
Coronavirus (2019-nCoV)		Market-Macro
Corporate Taxes		Market-Macro
Economic Conditions and Trends		Market-Macro
Elections		Market-Macro

Fig 3: Rules based classification pipeline for News Articles

3.2.2 NLP Sentiment Analysis

- Main and Abstract were identified to be the most suitable for sentiment analysis – better quality and was capturing the article's sentiment better. Main was also identified to be the headline in the URLs.
- TextBlob Sentiment Analysis: Applied TextBlob to compute polarity and subjectivity for different text columns.
- FinBERT Sentiment Analysis: Used a pre-trained FinBERT model to analyse sentiment specifically for financial news. Composite score was generated as Positive – Negative value. The composite score was generated on Main and substituted with Abstract's score if Main was empty.

$$\boldsymbol{Sentiment\ Score(Article) = FinBERT\_Score(Main)\ or\ FinBERT\_Score(Abstract)}$$

- Combined TextBlob (Abstract)and FinBERT (Main or Abstract) scores to create a weighted average sentiment score.
- FinBERT Sentiment score was confirmed to capture the sentiment better than TextBlob or the Weighted Average and retained as the final sentiment score.

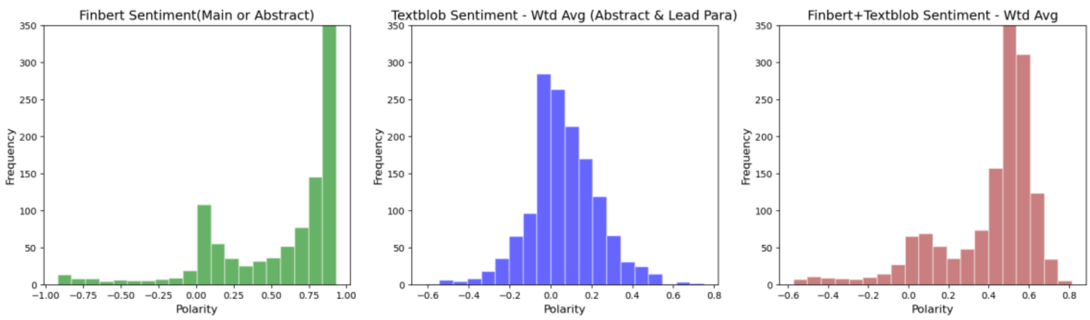


Fig 4: Comparative histogram of Sentiment score distributions

3.2.3 Stock and Sentiment Data processing

- Stock Data Load: Stock data was loaded from a pre-processed CSV file containing OHLCV data and earnings flags.
- Articles were categorized into company-specific and macro-level sentiments.
- Articles published on non-trading days were adjusted to the next trading day, ensuring sentiment is aligned to actual trading days, avoiding gaps in data.

3.2.4 Exploratory Data Analysis

- Analysed the impact of sentiment on stock returns around news events. A pre/post 5-day window was chosen to capture immediate market reactions.
- Grouped sentiment into positive, neutral, and negative to compare their effects on stock performance.

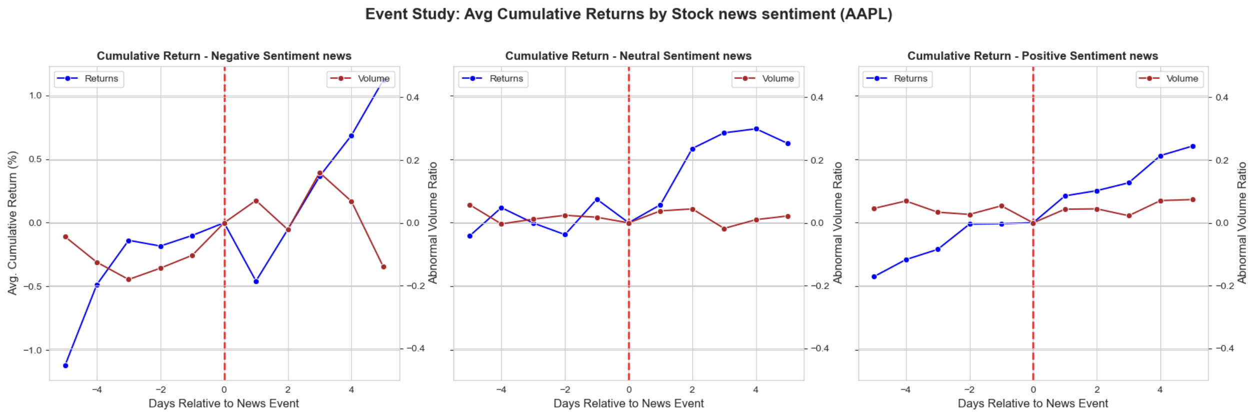


Fig 5: Example of Event Study Analysis chart - Apple for Stock sentiment

- Earnings Impact Analysis: Analysed the impact of sentiment on stock performance around earnings announcements. Created detailed plots showing pre and post-earnings sentiment trends, their correlation with stock returns& volume.

3.2.5 Modelling and Prediction

- Linear Regression, Random Forest and XGBoost on post announcement returns vs Company and Macro sentiment was evaluated. Directional Accuracy was observed to be a better metric than the prediction of returns.
- **Binary Logistic and Multinomial Regression** was evaluated – with the predicted variable being the direction of post returns (window of 1-3 days).
- The Multinomial Logistic regression model was overfitting for the directional predictions, but getting only the non-significant values correctly, Binary Logistic Regression model was underfitting as the sample size was very less for significant values.

Table 1: Multinomial vs Binary Logistic Regression for Sentiment based Prediction - Results

Quartile value Prediction
(Multinomial Logistic Regression)

Y: if 1 day post return < -0.96% assign -1, >1.68% assign 1, else assign 0.
Predict Y value (-1/0/1) based on Company and Macro Sentiment
Note:
- *Y was trialled for 1-3 day window*
- *Thresholds calculated as 0.25 and 0.75 quantiles*

	Shift	Train_Accuracy	Test_Accuracy	N_Observations
0	1	90.40%	91.06%	1510
1	2	85.10%	80.13%	1510
2	3	84.19%	79.14%	1510

Binary value prediction
(Binary Logistic Regression)

Y: if 1 day post return < -1.22% or >+1.22% assign 1, else assign -1.
Predict Y value (-1/1) based on Company and Macro Sentiment
Note:
- *Y was trialled for 1-3 day window*
- *Threshold calculated as std-Dev/2 of returns.*

	Shift	Train_Accuracy	Test_Accuracy	N_Observations
0	1	56.83%	65.71%	174
1	2	62.30%	62.50%	239
2	3	62.26%	62.26%	265

- Prediction models were mostly ineffective in predicting returns directly but were better at predicting the direction of movement.

Sentiment based Insights:

Apple	• Shows consistent recovery from negative news with positive drift otherwise • Pre-earnings optimism typically followed by post-earnings sentiment decline
Google	• Demonstrates high stability against negative news • Shows counter-intuitive positive returns on negative macro news • Pre-earnings optimism followed by post-earnings decline
NVIDIA	• Highly volatile response to news sentiment • Limited earnings-related sentiment data affects reliability of analysis

3.3 Earnings and Stock Trend Analysis

Python Notebook to refer: **EDA_Macros_Earnings_Team1_VP.ipynb**

The relationship between earnings-related data and post-earnings announcement stock returns for Apple (AAPL), Google (GOOGL), and Nvidia (NVDA). The objective is to understand the impact of earnings announcements on short-term stock price movements.

3.3.1 Data and Methodology

- Dataset covers AAPL, GOOGL, NVDA historical earnings data and stock price data.
- Metrics include EPS/revenue surprises and pre/post announcement returns.

Key Decisions

- The dataset is pre-processed to address potential issues such as outliers and skewness.
- Winsorization is applied to cap extreme values in the dependent variable (post-earnings announcement returns).
- A log(1 + y) transformation is used to mitigate the impact of skewness in the dependent variable, improving the normality of its distribution.
- EDA techniques are employed to understand the characteristics of the data.
- Distributions and descriptive statistics of key variables are examined to identify patterns, potential problems, and the need for further data transformations.

3.3.2 Results

Correlation Analysis

- A notable finding from the correlation analysis is that **EPS Surprise (%) and Surprise_Score** show a positive correlation with post-earnings announcement returns, particularly for the 1-day and 2-day windows. This suggests that companies that exceed earnings expectations tend to experience positive stock price reactions.

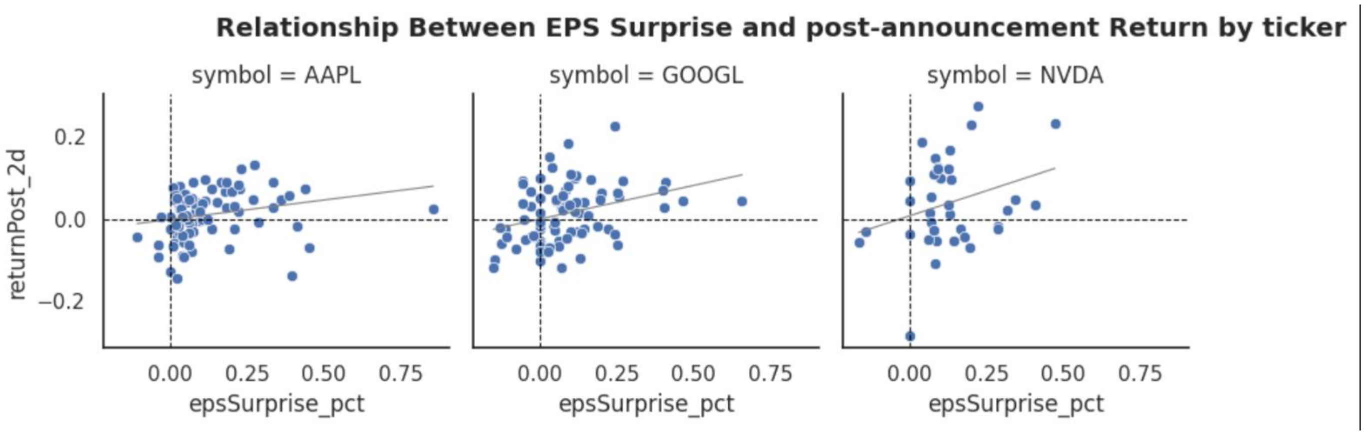


Fig 6 : EPS Surprise % Vs Post 2day Stock Returns - Scatterplot

- The **Balance of Power (BOP) indicator**, which captures **market momentum**, also shows some positive correlation with post-earnings announcement returns, especially for shorter return windows.
- Time-Varying Relationships: The correlation analysis conducted over different time periods (e.g., 2000-2004, 2005-2010, etc.) reveals that the strength and direction of these relationships can vary over time.

Insight:

- BOP relationship indicates companies with positive momentum leading up to earnings announcements are more likely to experience positive stock price movements after the announcement.
- Time varying relationship indicates the dynamic nature of market and the need for continuous monitoring and adaptation of analysis strategies.

Feature Analysis

- **Key Predictive Features:** Based on the correlation analysis and the nature of the data, the following features emerge as potentially important predictors of post-earnings announcement returns:
  - EPS Surprise (%)
  - Surprise_Score
  - Balance of Power (BOP) using Simple Moving Average (SMA)
  - Revenue Surprise (%) (though the correlation is generally weaker compared to EPS surprise)
- **Company-Specific Considerations:** The relative importance of these features may vary across different companies. For example, EPS surprise might be a stronger predictor for AAPL, while market momentum (BOP) could be more influential for GOOGL.

3.4 Macro-economic Factors Analysis

Python Notebook to refer: **EDA_Macros_Earnings_Team1_VP.ipynb**

The relationship between macroeconomic indicators and stock market performance for AAPL, GOOGL, and NVDA was analysed. It focuses on understanding how macroeconomic factors like Real GDP growth, Federal Debt, inflation (CPI), Unemployment, and US Treasury yields correlate with the companies' revenue growth and stock returns.

3.4.1 Data and Methodology

- The analysis utilizes the macro_metrics_dataset.csv, which contains historical macroeconomic and company-specific data. The dataset spans the period from 2000 to 2024.

Key Decisions

- **Correlation Analysis:** Correlation matrices and heatmaps are employed to visually identify relationships between macroeconomic factors and the companies' stock returns and revenue growth.
- **Regression Analysis:** Ordinary Least Squares (OLS) regression models are built to investigate the predictive capabilities of macroeconomic indicators on company revenue changes.
- **Time Series Analysis:** The relationship between GDP growth and revenue growth is analysed over different time periods (e.g., 2000-2004, 2005-2010) to identify evolving trends.

3.4.2 Results

Correlation Analysis

Individual Company Correlations (Fig 8): Heatmaps revealed notable differences in correlations between macro indicators and each company's performance.

- Apple: Strong correlation with GDP Growth, Federal Debt
- Google: Strong correlation with GDP Growth, CPI
- Nvidia: Strong correlation with GDP Growth, Treasury Yields

Combined Correlations (Fig 9): Visualizing correlations across all three companies highlighted the unique relationships between macroeconomic factors and stock returns. **This underscores how underlying business models and market segments can influence their performance.**

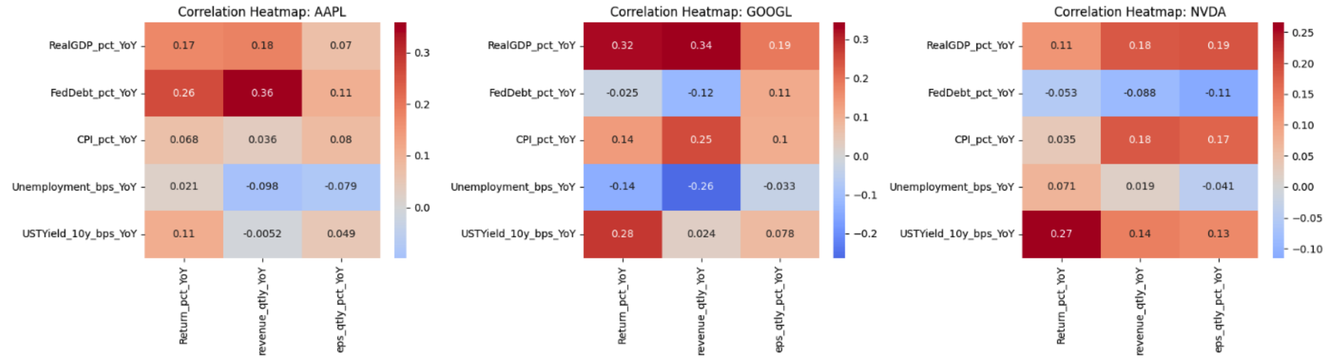


Fig 7 : Macro vs Company wise Correlation Heatmap

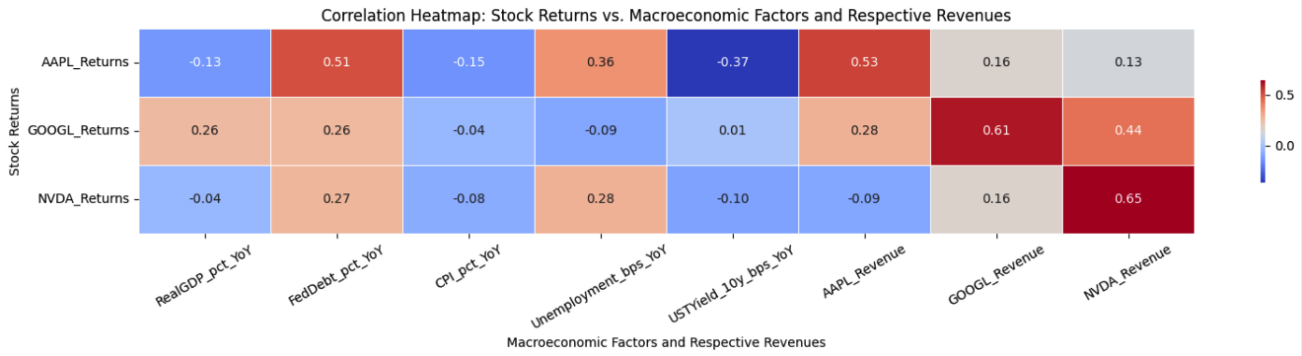
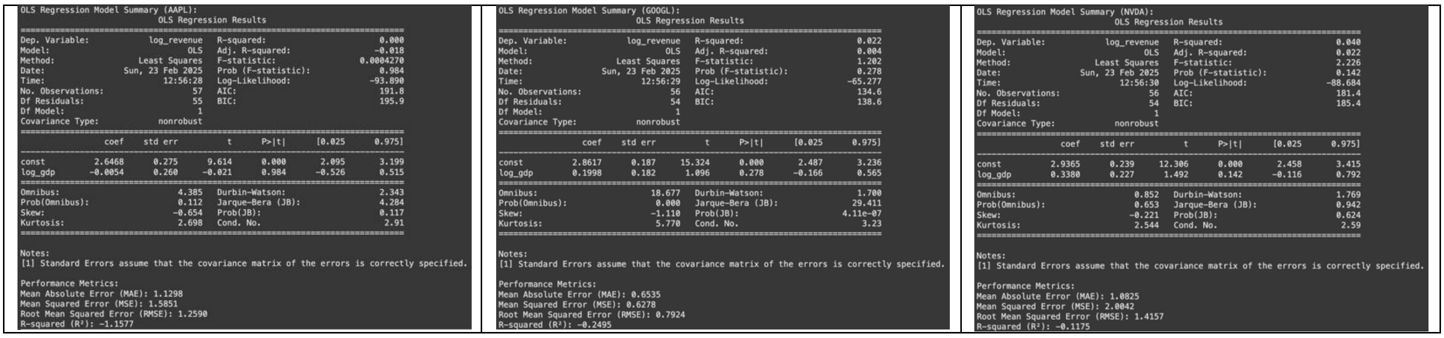


Fig 8: Combined Correlation Heatmap - Returns vs Revenue and Macro factors

Regression Analysis

- OLS regression models were built to predict revenue growth using macroeconomic indicators. The results with negative R^2 indicates that macroeconomic indicators alone do not reliably predict future revenue growth.



```
OLS Regression Model Summary (AAPL):
                            OLS Regression Results
==============================================================================
Dep. Variable:            log_revenue   R-squared:                       0.000
Model:                            OLS   Adj. R-squared:                 -0.018
Method:                 Least Squares   F-statistic:                 0.0004270
Date:                Sun, 23 Feb 2025   Prob (F-statistic):              0.984
Time:                        12:56:28   Log-Likelihood:                -93.890
No. Observations:                  57   AIC:                             191.8
Df Residuals:                      55   BIC:                             195.9
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          2.6468      0.275      9.614      0.000       2.095       3.199
log_gdp       -0.0054      0.260     -0.021      0.984      -0.526       0.515
==============================================================================
Omnibus:                        4.385   Durbin-Watson:                   2.343
Prob(Omnibus):                  0.112   Jarque-Bera (JB):                4.284
Skew:                          -0.654   Prob(JB):                        0.117
Kurtosis:                       2.698   Cond. No.                         2.91
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Performance Metrics:
Mean Absolute Error (MAE): 1.1298
Mean Squared Error (MSE): 1.5851
Root Mean Squared Error (RMSE): 1.2590
R-squared (R²): -1.1577
```

```
OLS Regression Model Summary (GOOGL):
                            OLS Regression Results
==============================================================================
Dep. Variable:            log_revenue   R-squared:                       0.022
Model:                            OLS   Adj. R-squared:                  0.004
Method:                 Least Squares   F-statistic:                     1.202
Date:                Sun, 23 Feb 2025   Prob (F-statistic):              0.278
Time:                        12:56:29   Log-Likelihood:                -65.277
No. Observations:                  56   AIC:                             134.6
Df Residuals:                      54   BIC:                             138.6
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          2.8617      0.187     15.324      0.000       2.487       3.236
log_gdp        0.1998      0.182      1.096      0.278      -0.166       0.565
==============================================================================
Omnibus:                       18.677   Durbin-Watson:                   1.700
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               29.411
Skew:                          -1.110   Prob(JB):                     4.11e-07
Kurtosis:                       5.770   Cond. No.                         3.23
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Performance Metrics:
Mean Absolute Error (MAE): 0.6535
Mean Squared Error (MSE): 0.6278
Root Mean Squared Error (RMSE): 0.7924
R-squared (R²): -0.2495
```

```
OLS Regression Model Summary (NVDA):
                            OLS Regression Results
==============================================================================
Dep. Variable:            log_revenue   R-squared:                       0.040
Model:                            OLS   Adj. R-squared:                  0.022
Method:                 Least Squares   F-statistic:                     2.226
Date:                Sun, 23 Feb 2025   Prob (F-statistic):              0.142
Time:                        12:56:30   Log-Likelihood:                -88.684
No. Observations:                  56   AIC:                             181.4
Df Residuals:                      54   BIC:                             185.4
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          2.9365      0.239     12.306      0.000       2.458       3.415
log_gdp        0.3380      0.227      1.492      0.142      -0.116       0.792
==============================================================================
Omnibus:                        0.852   Durbin-Watson:                   1.769
Prob(Omnibus):                  0.653   Jarque-Bera (JB):                0.942
Skew:                          -0.221   Prob(JB):                        0.624
Kurtosis:                       2.544   Cond. No.                         2.59
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Performance Metrics:
Mean Absolute Error (MAE): 1.0825
Mean Squared Error (MSE): 2.0042
Root Mean Squared Error (RMSE): 1.4157
R-squared (R²): -0.1175
```

Fig 9 : OLS Regression Results - Company Revenue vs GDP

Time Series Analysis

Comparing correlations between GDP growth and revenue growth over different periods revealed the evolving dependence of companies on macroeconomic conditions. For instance, Google's revenue growth was found to align more closely with GDP growth, while Apple and Nvidia's revenue displayed more independence.

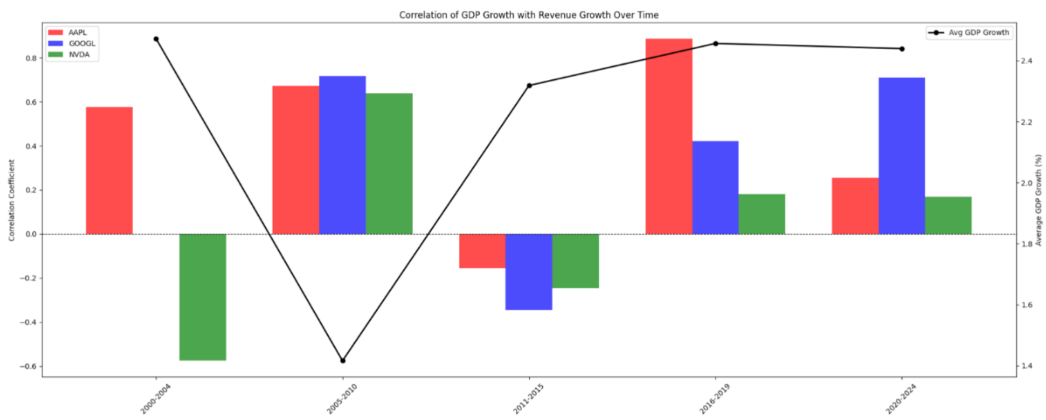


Fig 10: Correlation of GDP Growth with Revenue Growth over time

3.5 Trading Strategy

Python Notebook to refer: **Trading_strategy_predictive_modelling_Team1_VP.ipynb**

An ML trading strategy using **Random Forest Classifier** was developed for AAPL, GOOGL, and NVDA stocks. The model generates Long/Short/Hold signals based on price momentum, financial, and macro indicators. Initial results show promise, though further validation across market conditions is recommended.

3.5.1 Data Source and Description

The strategy utilizes the "strategy_dataset_export_wsenti.csv" dataset from the data sourcing notebook and contains stock price momentum, financial, macroeconomic, and sentiment analysis features for AAPL, GOOGL, NVDA

3.5.2 Data Preprocessing and Feature Engineering

Key Decisions

- Missing values in sentiment columns are filled with 0.
- Feature Subsetting: Relevant features were selected for model training.
- The target variable **'signal'** is calculated based on post-signal returns (2-day or 5-day holding periods) using predefined thresholds (hold_max, hold_min). Thresholds were set to ±5% initially and later adjusted to ±2% for AAPL. For GOOGL and NVDA the thresholds are set to ±5%
- Data was split into training (90%) and testing (10%) sets using stratified sampling to maintain class distribution.

3.5.3 Model Development and Hyperparameter Tuning

A **Random Forest Classifier** was employed as the prediction model.

- Baseline Model: A baseline model was trained with default hyperparameters.
- Hyperparameter Tuning: **RandomizedSearchCV** is utilized to optimize hyperparameters including n_estimators, max_depth, min_samples_split, min_samples_leaf, and max_features, using 'precision_weighted' as the scoring metric.
- Final Model: The best hyperparameters were used to train the final model for each stock.

3.5.4 Trading Strategy Simulation and Evaluation

The strategy simulation generates trading signals based on the final model's predictions and backtesting on historical data. Performance evaluation includes calculating:

Evaluation Metric	Definition
Cumulative Return	Overall return of the strategy over time for each stock
Drawdown	Maximum percentage decline from a peak to a trough in cumulative return for each stock.
Average Return per Trade	Mean return per trade executed for each stock.
Win Rate	Percentage of profitable trades for each stock.
Precision	The proportion of correctly predicted signals out of all predicted signals for each stock.

3.5.5 Results

3.5.5.1 Model Performance

Across all three stocks, both baseline and tuned models demonstrated relatively high precision across training and testing datasets, indicating the model's ability to effectively identify trading signals with minimal false positives. The tuned models exhibited better generalization, mitigating potential overfitting observed in the baseline models.

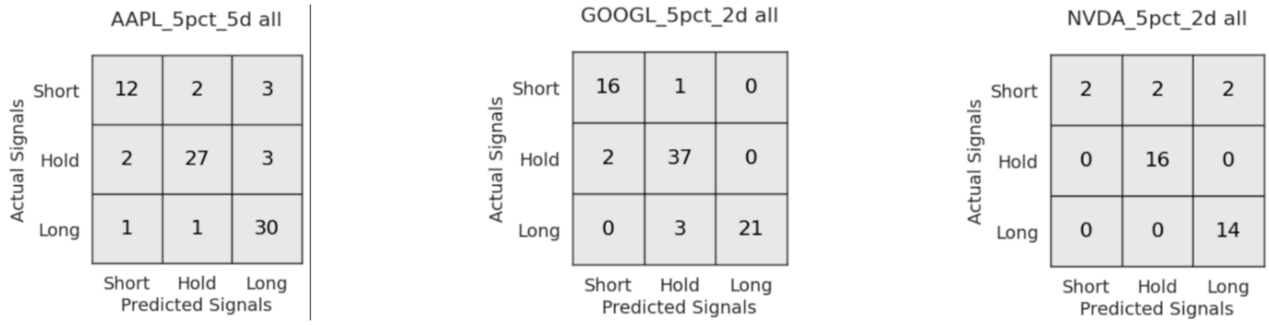


Fig 11: Trading RF Prediction Model - Company-wise Confusion Matrix

3.5.5.2 Strategy Performance

The strategy, when applied to AAPL, GOOGL, and NVDA, yields varying cumulative returns over the historical period. AAPL and NVDA show positive cumulative returns, while GOOGL's performance is more sensitive to model parameters and thresholds.

Insight: These performance variations highlight the importance of model customization and parameter optimization for different stocks and market conditions.

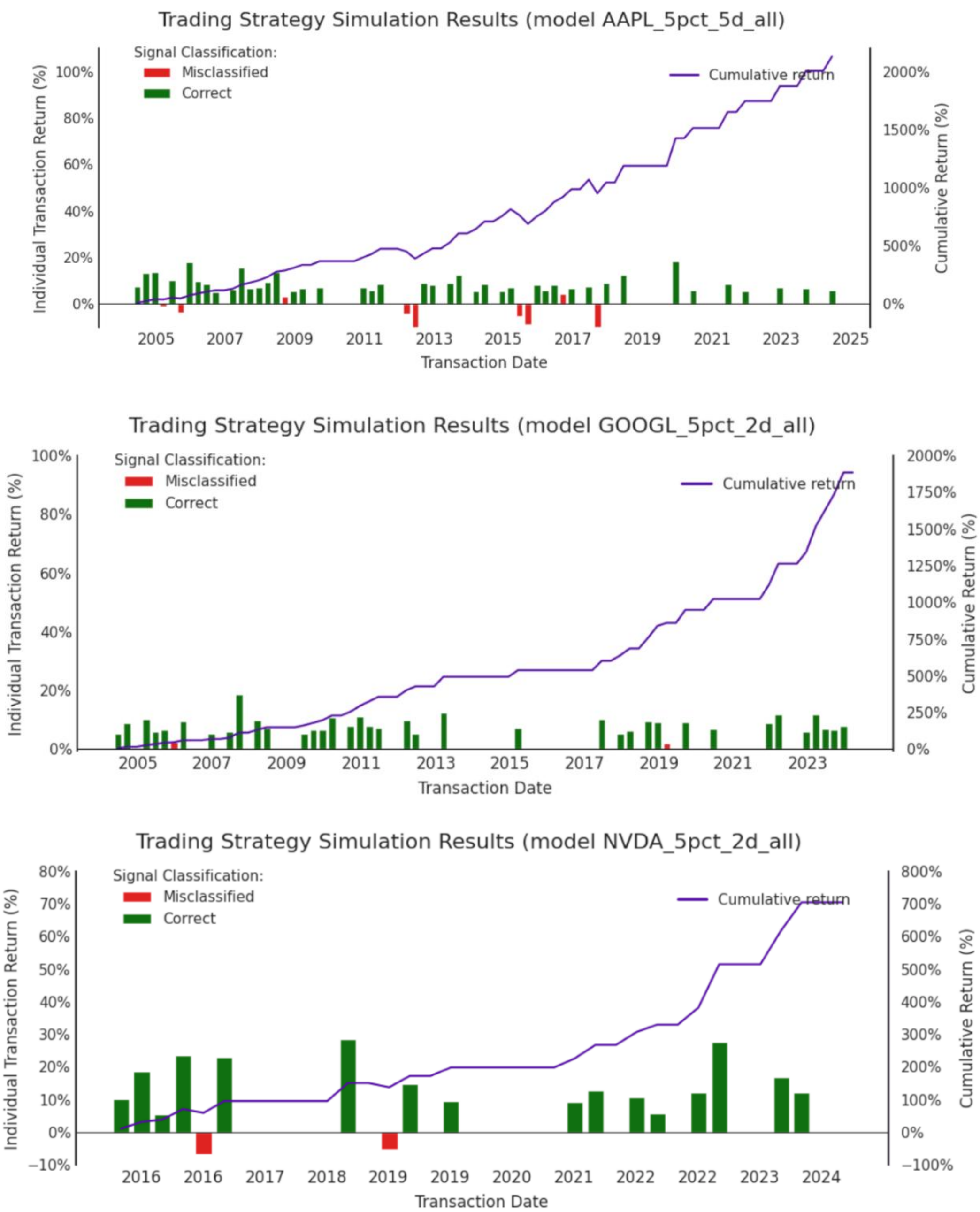


Figure 12 : Trading Strategy Backtesting Simulation Result

3.5.5.3 Feature Importance Analysis

Key Insight

- Macroeconomic and sentiment features play a less prominent role in prediction compared to other financial indicators.
- This finding suggests potential areas for refinement in feature engineering and selection.

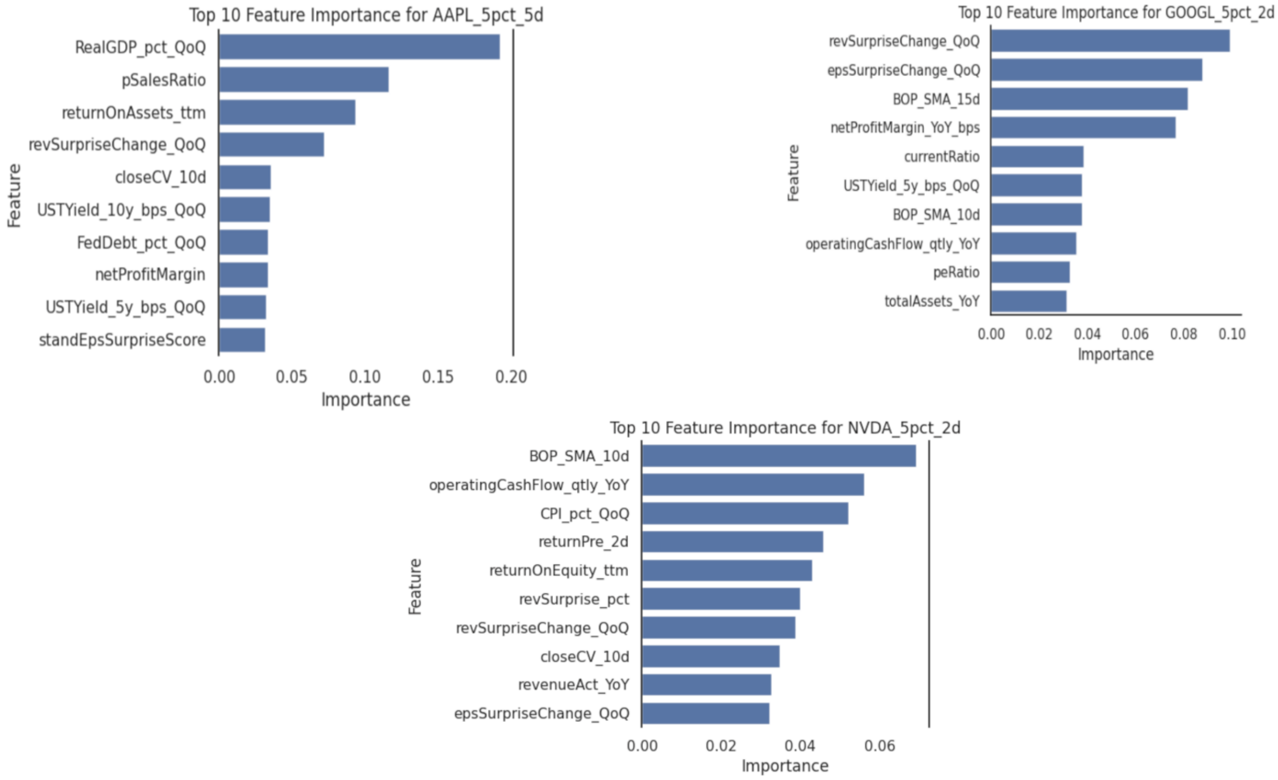


Figure 13: Top 10 Features for Company wise models

4. Recommendations

4.1 Business Recommendations

- **Stock-Specific Approach**: Develop customized trading strategies per stock - AAPL showed stronger debt correlations, GOOGL with CPI, and NVDA with treasury yields
- **Earnings Strategy:** Focus on EPS surprise and BOP (Balance of Power) indicators around earnings, as they showed strongest correlation with post-announcement returns
- **Risk Management:** Implement different thresholds for different stocks based on their volatility patterns (e.g., tighter bounds for AAPL vs NVDA)
- **Sentiment Integration:** Use sentiment analysis as a supplementary rather than primary signal, given its limited predictive power in isolation
- **Time Window Optimization:** Adjust trading windows based on stock-specific behaviour patterns (e.g., 2-day vs 5-day holding periods showed varying effectiveness)

4.2 Technical Recommendations

- **Model Validation and Robustness:** Conduct comprehensive out-of-sample testing and backtesting across diverse market conditions and a broader range of assets to ensure robustness and generalizability.
- **Feature Engineering and Selection:** Explore and incorporate additional relevant features tailored to each stock to further enhance predictive power and capture stock-specific characteristics.
- **Hyperparameter Optimization:** Fine-tune hyperparameters for each stock and specific model settings to optimize performance and adapt to individual stock behaviour.
- **Model Monitoring and Maintenance:** Establish a system for continuous model updates and performance monitoring to adapt to evolving market dynamics and ensure the strategies' long-term effectiveness.

Appendix 1 – References

- https://www.lgtcp.com/files/2024-04/lgt_capital_partners_-_hedge_fund_strategies_introduction_-_2024_en.pdf
- https://wandb.ai/ivangoncharov/FinBERT_Sentiment_Analysis_Project/reports/Financial-Sentiment-Analysis-on-Stock-Market-Headlines-With-FinBERT-HuggingFace--VmlldzoxMDQ4NjM0
- https://site.financialmodelingprep.com/developer/docs/stable#key-metrics
- https://www.sec.gov/cgi-bin/browse-edgar?action=getcompany&CIK=0001350694&owner=exclude&count=40&hidefilings=0
- https://blog.quantinsti.com/stock-market-data-analysis-python/
- https://macrosynergy.com/research/six-ways-to-estimate-realized-volatility/
- https://www.alphavantage.co/documentation/
- https://github.com/RomelTorres/alpha_vantage
- https://yfinance-python.org/
- https://quantpedia.com/strategies/post-earnings-announcement-effect/
- https://www.eventstudytools.com/introduction-event-study-methodology
- https://mba.tuck.dartmouth.edu/pages/faculty/ken.french/Data_Library/f-f_factors.html
- https://www.ravenpack.com/research/uncover-alpha-with-earnings-calls-transcript-data?utm_source=youtube&utm_medium=video&utm_campaign=3tradingsignals

Appendix 2 – Derived Features Data Dictionary

Feature name	Formula
dailyReturn	Daily Return (%) = (Daily Closing Price[t] / Daily Closing Price[t - 1]) - 1
BOP	Balance of Power (BOP) = (Daily Closing Price - Daily Opening Price) / (Daily Highest Price - Daily Lowest Price)
returnPre_1d,2d,3d,4d,5d,10d,15d,20d	Pre-announcement Return (X Day) = (Daily Closing Price[t] / Daily Closing Price[t - X]) - 1
closeCV_2d,3d,4d,5d,10d,15d	Closing Price Coefficient of Variation (2 Days) = Standard deviation of Daily Closing Price over the last X days / Average Daily Closing Priceover the last X days
BOP_SMA_2d,3d,4d,5d,10d,15d	X-day average of Balance of Power (BOP)
returnPost_1d,2d,3d,4d,5d,10d,15d,20d	Post-announcement Return (1 Day) = (Daily Closing Price[t + X] / Daily Closing Price[t]) - 1
RealGDP_pct_QoQ	Real GDP Growth QoQ, % = (Real Gross Domestic Product (GDPC1)[t] / Real Gross Domestic Product (GDPC1)[t - 1]) - 1, where t represents the current quarter. t−1 represents the previous quarter
RealGDP_pct_YoY	Real GDP Growth YoY, % = (Real Gross Domestic Product (GDPC1)[t] / Real Gross Domestic Product (GDPC1)[t - 4]) - 1, where t represents the current quarter. t−4 represents the same quarter in the previous year
FedDebt_pct_QoQ	Federal Debt Growth QoQ, % = (Federal Debt (GFDEBTN)[t] / Federal Debt (GFDEBTN)[t - 1]) - 1, where t represents the current quarter. t−1 represents the previous quarter
FedDebt_pct_YoY	Federal Debt Growth YoY, % = (Federal Debt (GFDEBTN)[t] / Federal Debt (GFDEBTN)[t - 4]) - 1, where t represents the current quarter. t−4 represents the same quarter in the previous year
CPI_pct_QoQ	Inflation QoQ, % = (Consumer Price Index (CPIAUCSL)[t] / Consumer Price Index (CPIAUCSL)[t - 1]) - 1, where t represents the current quarter. t−1 represents the previous quarter
CPI_pct_YoY	Inflation YoY, % = (Consumer Price Index (CPIAUCSL)[t] / Consumer Price Index (CPIAUCSL)[t - 4]) - 1, where t represents the current quarter. t−4 represents the same quarter in the previous year
Unemployment_bps_YoY	Unemployment Change YoY, bps = Unemployment Rate (UNRATE)[t] - Unemployment Rate (UNRATE)[t - 4], where t represents the current quarter. t−4 represents the same quarter in the previous year
USTYield_10y_bps_YoY	US Treasury Yield Change 10Y YoY, bps = US Treasury Yield (10Y)[t] - US Treasury Yield (10Y)[t - 4], where t represents the current quarter. t−4 represents the same quarter in the previous year
USTYield_5y_bps_YoY	US Treasury Yield Change 5Y YoY, bps = US Treasury Yield (5Y)[t] - US Treasury Yield (5Y)[t - 4], where t represents the current quarter.t−4 represents the same quarter in the previous year
USTYield_2y_bps_YoY	US Treasury Yield Change 2Y YoY, bps = US Treasury Yield (2Y)[t] - US Treasury Yield (2Y)[t - 4], where t represents the current quarter. t−4 represents the same quarter in the previous year

USTYield_3m_bps_YoY	US Treasury Yield Change 3M YoY, bps = US Treasury Yield (3M)[t] - US Treasury Yield (3M)[t - 4], where t represents the current quarter. t−4 represents the same quarter in the previous year
Unemployment_bps_QoQ	Unemployment Change QoQ, bps = Unemployment Rate (UNRATE)[t] - Unemployment Rate (UNRATE)[t - 1], where t represents the current quarter. t−1 represents the previous quarter
USTYield_10y_bps_QoQ	US Treasury Yield Change 10Y QoQ, bps = US Treasury Yield (10Y)[t] - US Treasury Yield (10Y)[t - 1], where t represents the current quarter. t−1 represents the previous quarter
USTYield_5y_bps_QoQ	US Treasury Yield Change 5Y QoQ, bps = US Treasury Yield (5Y)[t] - US Treasury Yield (5Y)[t - 1], where t represents the current quarter. t−1 represents the previous quarter
USTYield_2y_bps_QoQ	US Treasury Yield Change 2Y QoQ, bps = US Treasury Yield (2Y)[t] - US Treasury Yield (2Y)[t - 1], where t represents the current quarter. t−1 represents the previous quarter
USTYield_3m_bps_QoQ	US Treasury Yield Change 3M QoQ, bps = US Treasury Yield (3M)[t] - US Treasury Yield (3M)[t - 1], where t represents the current quarter. t−1 represents the previous quarter
grossProfitMargin_ttm	Gross Profit Margin (TTM) = Gross Profit, TTM / Revenue (TTM)
operatingProfitMargin_ttm	Operating Profit Margin (TTM) = Operating Income TTM / Revenue (TTM)
netProfitMargin_ttm	Net Profit Margin (TTM) = Net Income (TTM) / Revenue (TTM)
returnOnAssets_ttm	Return on Assets (TTM) = Net Income (TTM) / Total Assets
returnOnEquity_ttm	Return on Equity (TTM) = Net Income (TTM) / Total Stockholders' Equity
operatingCashFlowSalesRatio_ttm	Operating Cash Flow to Sales Ratio (TTM) = Operating Cash Flow TTM / Revenue (TTM)
dividendPaidAndCapexCoverageRatio_ttm	Dividend & CapEx Coverage Ratio (TTM) = Net Cash Provided by Operating Activities, TTM / Dividends Paid, TTM
dividendPayoutRatio_ttm	Dividend Payout Ratio (TTM) = Dividends Paid, TTM / Net Income (TTM)
revenue_ttm	Trailing twelve month revenue - to adjust seasonality
netIncome_ttm	Trailing twelve month Net Income - to adjust seasonality
currentRatio_YoY_bps	Current Ratio Change YoY, bps = Current Ratio[t] - Current Ratio[t - 4], where t represents the current quarter. t−4 represents the same quarter in the previous year
debtToAssets_YoY_bps	Debt to Assets Change YoY, bps = Debt to Assets[t] - Debt to Assets[t - 4], where t represents the current quarter. t−4 represents the same quarter in the previous year
debtToEquity_YoY_bps	Debt to Equity Change YoY, bps = Debt to Equity[t] - Debt to Equity[t - 4], where t represents the current quarter. t−4 represents the same quarter in the previous year
grossProfitMargin_ttm_YoY_bps	Gross Profit Margin Change YoY, bps = Gross Profit Margin (TTM)[t] - Gross Profit Margin (TTM)[t - 4], where t represents the current quarter. t−4 represents the same quarter in the previous year
operatingProfitMargin_ttm_YoY_bps	Operating Profit Margin Change YoY, bps = Operating Profit Margin (TTM)[t] - Operating Profit Margin (TTM)[t - 4], where t represents the current quarter. t−4 represents the same quarter in the previous year
netProfitMargin_ttm_YoY_bps	Net Profit Margin Change YoY, bps = Net Profit Margin (TTM)[t] - Net Profit Margin (TTM)[t - 4], where t represents the current quarter. t−4 represents the same quarter in the previous year
returnOnAssets_ttm_YoY_bps	Return on Assets Change YoY, bps = Return on Assets (TTM)[t] - Return on Assets (TTM)[t - 4], where t represents the current quarter. t−4 represents the same quarter in the previous year
returnOnEquity_ttm_YoY_bps	Return on Equity Change YoY, bps = Return on Equity (TTM)[t] - Return on Equity (TTM)[t - 4], where t represents the current quarter. t−4 represents the same quarter in the previous year
operatingCashFlowSalesRatio_ttm_YoY_bps	Operating Cash Flow to Sales Ratio Change YoY, bps = Operating Cash Flow to Sales Ratio (TTM)[t] - Operating Cash Flow to Sales Ratio (TTM)[t - 4], where t represents the current quarter. t−4 represents the same quarter in the previous year
dividendPaidAndCapexCoverageRatio_ttm_YoY_bps	Dividend & CapEx Coverage Ratio Change YoY, bps = Dividend & CapEx Coverage Ratio (TTM)[t] - Dividend & CapEx Coverage Ratio (TTM)[t - 4], where t represents the current quarter. t−4 represents the same quarter in the previous year

dividendPayoutRatio_ttm_YoY_bps	Dividend Payout Ratio Change YoY, bps = Dividend Payout Ratio (TTM)[t] - Dividend Payout Ratio (TTM)[t - 4], where t represents the current quarter. t−4 represents the same quarter in the previous year
totalAssets_YoY	Total Assets Change YoY, % = (Total Assets[t] / Total Assets[t - 4]) - 1, where t represents the current quarter. t−4 represents the same quarter in the previous year
revenue_qtly_YoY	Quarterly Revenue Change YoY, % = (Reported Revenue[t] / Reported Revenue[t - 4]) - 1, where t represents the current quarter.t−4 represents the same quarter in the previous year
operatingCashFlow_qtly_YoY	Quarterly Operating Cash Flow Change YoY, % = (Operating Cash Flow[t] / Operating Cash Flow[t - 4]) - 1, where t represents the current quarter. t−4 represents the same quarter in the previous year
revenue_ttm_YoY	TTM Revenue YoY, % = (Revenue (TTM)[t] / Revenue (TTM)[t - 4]) - 1, where t represents the current quarter. t−4 represents the same quarter in the previous year
operatingCashFlow_ttm_YoY	TTM Operating Cash Flow Change YoY, % = (Operating Cash Flow TTM[t] / Operating Cash Flow TTM[t - 4]) - 1, where t represents the current quarter. t−4 represents the same quarter in the previous year
netProfitMargin_YoY_bps	Net Profit Margin Change YoY, bps = Net Profit Margin[t] - Net Profit Margin[t - 4], where t represents the current quarter. t−4 represents the same quarter in the previous year
peRatio	Price-to-Earnings Ratio = Market Capitalisation / Net Income (TTM)
pSalesRatio	Price-to-Sales Ratio = Market Capitalisation / Revenue (TTM)
amcEquivalentEarnReportedDate	If earnReportedTime = 'amc', amcEquivalentEarnReportedDate = EarnReportedDate, other mcEquivalentEarnReportedDate = EarnReportedDate - 1
epsSurpriseAbs	EPS Surprise (Absolute) = EPS Actual - EPS Estimate
epsSurprise_pct	EPS Surprise (%) = EPS Surprise (Absolute) / EPS Estimate - 1
revSurpriseAbs	Revenue Surprise (Absolute) = Revenue Actual - Revenue Estimate
revSurprise_pct	Revenue Surprise (%) = Revenue Surprise (Absolute) / Revenue Estimate - 1
standEpsSurpriseScore	Standardised EPS Surprise Score = EPS Surprise (Absolute) / Standard deviation of EPS Surprise (Absolute) over the last 12 days
epsSurpriseChange_QoQ	EPS Surprise Change QoQ, bps = EPS Surprise (%)[t] - EPS Surprise (%)[t - 1], where t represents the current quarter. t−1 represents the previous quarter
epsEst_YoY	Quarterly EPS Estimate Change YoY, % = (EPS Estimate[t] / EPS Estimate[t - 4]) - 1, where t represents the current quarter. t−4 represents the same quarter in the previous year
epsAct_YoY	Quarterly EPS Actual Change YoY, % = (EPS Actual[t] / EPS Actual[t - 4]) - 1, where t represents the current quarter.t−4 represents the same quarter in the previous year
standRevSurpriseScore	Standardised Revenue Surprise Score = Revenue Surprise (Absolute) / Standard deviation of Revenue Surprise (Absolute) over the last 12 days
revSurpriseChange_QoQ	Revenue Surprise Change QoQ, bps = Revenue Surprise (%)[t] - Revenue Surprise (%)[t - 1], where t represents the current quarter. t−1 represents the previous quarter
revenueEst_YoY	Quarterly Revenue Estimate Change YoY, % = (Revenue Estimate[t] / Revenue Estimate[t - 4]) - 1, where t represents the current quarter. t−4 represents the same quarter in the previous year
revenueAct_YoY	Quarterly Revenue Actual Change YoY, % = (Revenue Actual[t] / Revenue Actual[t - 4]) - 1, where t represents the current quarter. t−4 represents the same quarter in the previous year