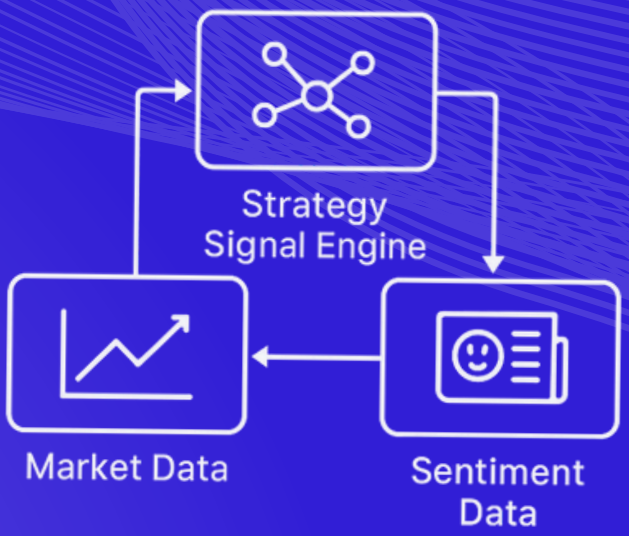
ZENTRYX

ZENTRYX

CRYPTO STRATEGY ENGINE

STAGE I & II REPORT – JULY 2025

PRESENTED TO : INSTITUTIONAL INVESTORS & STRATEGY LEADERS
PRESENTED BY : YIQING ZHU | UNSW FINTECH | JULY 2025



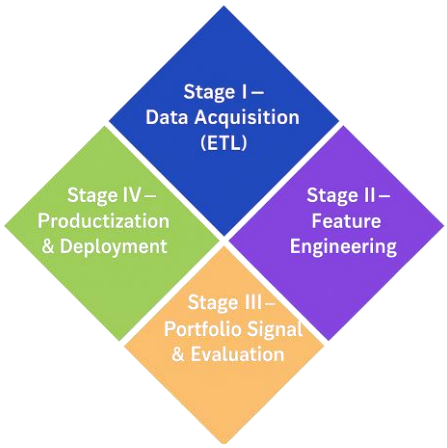
ZENTRYX

Executive Summary

OVERVIEW

Zentryx is a sentiment-enhanced cryptocurrency strategy engine designed to generate systematic weekly allocation signals for institutional investors.

Powered by a modular four-stage architecture—from data acquisition to live deployment, the engine systematically transforms structured market data and unstructured news sentiment into weekly allocation signals.



This report focuses on **Stage I (Data Acquisition)** and **Stage II (Feature Engineering)**, where both price-volume data and financial news are cleaned, filtered, and engineered into predictive inputs for downstream strategy design.

KEY FINDINGS AND RECOMMENDATIONS

Structured Data

Engineered predictive crypto factors (momentum, volatility, reversal) from clean OHLCV data; stablecoins and wrapped tokens excluded. Real-time WebSocket feed built for future live market response.

Unstructured Data

Developed a FinBERT+VADER hybrid sentiment pipeline with 0.73 F1-score and 96% positive recall: narrative-token links established via Granger tests.

Contents

ZENTRYX

STATION 1: ETL

1.1 Product Description

Zentryx is a cryptocurrency portfolio optimization system that integrates traditional technical indicators with sentiment-derived insights.

In Stage 1, Zentryx establishes a structured data pipeline to collect and clean daily OHLCV (open, high, low, close, volume) data for the top 200 cryptocurrencies, ranked by circulating market capitalization in USD. In addition, a real-time WebSocket pipeline has been developed to stream live cryptocurrency prices using the CryptoCompare API. For unstructured data, Zentryx also retrieves two years of historical crypto news from CoinDesk, enabling downstream sentiment analysis.

These datasets collected in Station 1 form the data lake foundation for feature engineering, historical backtesting, and real-time simulation—enabling robust portfolio signal generation in later stages.

1.2 Product Basic Use and Interface

Our product currently provides clients with real-time cryptocurrency price data through an interactive Streamlit-based front-end. The user interface features live candlestick charts, enriched with key technical indicators such as Exponential Moving Average (EMA) and Relative Strength Index (RSI). All streamed data is concurrently stored in CSV format for reproducibility and further analysis.

The real-time data feed forms a critical input for our short-term trading factor research. In Stage 3, this high-frequency data will support the development and backtesting of ultra-short-term alpha signals, aiming to capture intraday price inefficiencies.

For internal analytics, Zentryx also leverages historical OHLCV and news data to design and test medium- to long-term factor strategies. Specifically, historical price data is resampled to weekly frequency, reducing noise from daily volatility and creating a clean foundation for long-horizon factor modeling.

On the unstructured data side, we have extracted all CoinDesk news articles published between January 1, 2023, and July 21, 2025. These articles are systematically processed through a multi-stage natural language pipeline (detailed in Station #2), allowing us to quantify market sentiment and align it with asset-specific returns.

To ensure data quality and reliability, we adopt a hybrid API strategy:

- **GET requests** are used to retrieve structured historical price data and archived news content.
- **WebSocket connections** are deployed for real-time market data streaming, providing sub-second latency essential for short-term trading analytics.

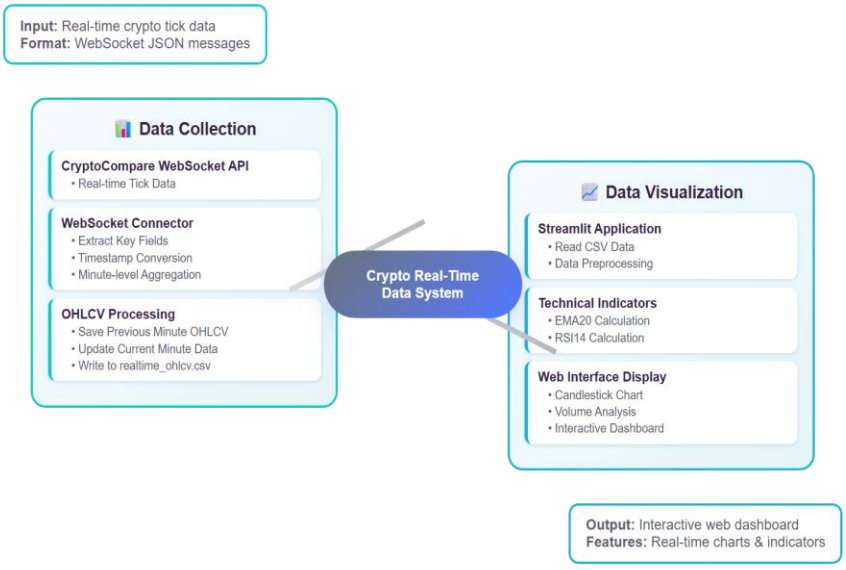
This dual-approach API architecture ensures that Zentryx is equipped to support a broad spectrum of trading styles — from high-frequency traders requiring instant market signals to institutional investors focused on long-term portfolio construction.

1.3 Data Processing

1.3.1 Structured Real-Time Data

Data Acquisition Module – realtime_ws_ohlcv.py

This module transforms high-frequency tick data from CryptoCompare's WebSocket stream into structured 1-minute OHLCV bars in real time. Each incoming tick (price, volume, timestamp) is dynamically assigned to its corresponding minute window, where the system tracks open, high, low, close, and volume cumulatively.



The aggregation mechanism ensures that:

- **Open** is the first trade of the minute,
- **High/Low** update with each new tick,
- **Close** reflects the latest price, and
- **Volume** is aggregated in real time.

To maintain robustness, the system includes:

- **Auto-reconnect logic** to recover from WebSocket drops
- **Fault-tolerant message handling**
- **Efficient memory control** using deque, which avoids memory overload while supporting real-time visualization.

At the start of each new minute, the complete OHLCV bar for the previous minute is immediately written to disk (CSV), and the system resets for the next interval. This rolling window ensures temporal accuracy and continuity. The sample output csv and Streamlit platform picture is in Appendix.

Table 1: Sample Real-Time BTC/USD 1-Min OHLCV Data

Symbol	Datetime	Open	High	Low	Close	USD Volume	BTC Volume	USD Vol (mil)
BTC	7/20/2025 8:34	118217.9	118219.9	118217.8	118219.9	2411.01	0.020394	0.002411
BTC	7/20/2025 8:35	118219.9	118221.7	118219.5	118221.7	24921.69	0.210805	0.024922
BTC	7/20/2025 8:36	118221.7	118224.6	118218.0	118218.0	17305.74	0.146388	0.017306
BTC	7/20/2025 8:37	118218.0	118222.7	118215.9	118217.8	5936.0	0.050212	0.005936
BTC	7/20/2025 8:38	118217.8	118253.1	118217.8	118253.1	35535.01	0.300500	0.035535
BTC	7/20/2025 8:39	118253.1	118274.1	118253.1	118274.1	75984.94	0.642448	0.075985

1.3.2 Structured Historical Data

Data Acquisition Module – crypto_pipeline.py

In this section, we filter the coins we want by market capitalization using the get_top_coins() function, which directly retrieves the top 200 coins at the time the data is downloaded. This ensures that none of the coins are delisted and that the investment universe consists of actively traded, high-liquidity assets. get_daily_ohlcv2() is a core function for fetching the complete data from API. The function attempts up to max_retries calls to the API. Retries are triggered under the following conditions:

- The API response explicitly returns an error (e.g., "Response": "Error")
- Critical fields such as TIMESTAMP are missing, indicating incomplete or malformed data
- Network-related exceptions occur, such as timeouts or connection failures (requests. Request Exception)

This ensures that the output data is complete and time-aligned, minimizing the risk of missing values that could affect downstream processing in Station 2. The output from Station 1 is in long format, where each row represents a single token–date combination and contains standardized OHLCV data, including both USD volume and BTC volume, with USD volume additionally expressed in millions.

1.3.3 Unstructured News Data

Data Acquisition Module - crypto_news_pipeline.py

In this section, stage1_load_news () did a main job on cleaning the text, it dropped the verbose and unrelated information from the raw data, such as subtitles, authors, URL, creating time etc. Stage 1 only keeps the date, id, published unique id, title, body, keywords, language, and sentiment (postiive:1, negative: 0) from the raw Json data and transforms it to a long data frame. Table 2: stage 1 new pipeline output

date	id	published_on	title	body	keywords	lang	positive
7/20/2025 14:00	48600435	1753020000	Block: Crypto Mi	Summary Block	XYZ	EN	0
7/20/2025 14:00	48601444	1753020000	Chart of the Wee	Wall Street has	Markets|mar	EN	0
7/20/2025 13:57	48600360	1753019855	Euro debt deals	Emerging-mark	Economy|EU	EN	1
7/20/2025 13:56	48600272	1753019803	XRP price predic	XRP is experien	Cryptocurren	EN	1
7/20/2025 13:54	48600147	1753019650	Bitcoin Price An	Bitcoin (BTC) h	Breaking Nev	EN	0
7/20/2025 13:51	48608392	1753019483	Best Crypto to B	The crypto mar	More News	EN	1
7/20/2025 13:44	48599729	1753019098	Hoskinson prom	Charles Hoskin	Cardano|AD/	EN	0

ZENTRYX

STATION 2
Feature Engineering

2.1 Structured Data Feature Engineering

The system This study employs four categories of technical indicators:

- Trend and Momentum indicators (RSI14, EMA, momentum_14, slope_14) - capture price direction and velocity
- Volatility indicators (volatility, strev_daily/weekly, vol_spike_rate_28d) - measure price uncertainty
- Risk-adjusted indicators (drawdown, alpha_vs_btc, trend_smoothness_14) - evaluate risk-return ratios
- Volume indicators (volume) - reflect market participation

2.1.1 Trend and Momentum indicators

RSI14: RSI = 100 - (100/ (1 + RS)) where RS = Average Gain/Average Loss over 14 days

EMA: $EMA_t = \alpha \times P_t + (1-\alpha) \times EMA_{t-1}$ with exponential decay factor $\alpha = 2/(N+1)$

Momentum_14: $(P_t - P_{t-14})/P_{t-14} \times 100$ measuring 14-day price rate of change

Slope_14: Linear regression slope of price over 14-day window

Momentum trading methodologies operate by establishing upper and lower boundary conditions within oscillating technical indicators to forecast trend direction changes. Victor Alexandre Padilha et al.'s research uses RSI, SRSI, and WR indicators where: (1) buy signals emerge when securities breach above the lower oversold threshold, potentially marking the transition from declining to ascending price trends, and (2) sell signals when securities fall below the upper overbought threshold, potentially signaling the shift from rising to declining price movements. (Victor Alexandre Padilha et al. 2024) Building on this framework, our system employs a comprehensive set of four technical indicators: RSI14, EMA, momentum_14, and slope_14 to test the momentum-based trading strategies.

2.1.2 Volatility indicators

Volatility = $\sqrt{}$ $(\sum(r_i - \bar{r})^2/(n-1)) \times \sqrt{252}$ where r_i = daily return, $\bar{r}$ = mean return, n = observation period, measuring annualized price uncertainty

STREV_daily_t = $-\sum(r_{t-i})$ for i = 1 to 5 where r_{t-i} = return at day t-i, measuring 5-day short-term reversal effect

STREV_weekly = $-\sum(rweek_{t-i})$ for i = 1 to 4 where $rweek_{t-i}$ = return at week t-i, measuring 4-week short-term reversal effect

Vol_spike_rate_28d = $(\sum Vol\_spike_i/28) \times 100$ where $Vol\_spike_i = 1$ if $Vol_i > (Vol\_MA_{20i} + 1.5 \times Vol\_STD_{20i})$, else 0, measuring frequency of volatility spikes over 28 days

Volatility exhibits strong correlations with price movements and market sentiment, particularly pronounced in cryptocurrency markets characterized by elevated volatility levels. The research by Muguto, H.T. et al. (2022) further confirms a positive relationship between investor sentiment and sector return volatilities. At Zentryx, we will investigate the cross-sectional dynamics between volatility and sentiment, exploring how these interconnected factors influence market behavior and can be leveraged for enhanced analytical insights.

2.1.3 Risk-adjusted indicators

Drawdown = (Peak Value - Trough Value) / Peak Value × 100% | Measuring maximum peak-to-trough decline to assess downside risk

Maximum Drawdown = max(DDt) where DDt = (Pt - max (P0, P1, ..., Pt)) / max(P0, P1, ..., Pt)

Alpha_vs_BTC = Rp - [Rf + β × (RBTC - Rf)] Where: β = Cov(Rp, RBTC) / Var(RBTC) Rp = Portfolio return RBTC = Bitcoin return Rf = Risk-free rate| Calculating excess returns above Bitcoin as a market benchmark

Trend_Smoothness_14 = 1 - (Σ|Rt - SMA14t|) / (14 × SMA14t) Evaluating price trajectory consistency over a 14-day window to gauge trend reliability.

2.1.4 Volume indicators

Volume indicators involve in volume_usd, volume_btc, volume_usd_mile that we output in Station 1 and a set of volume shock indicators using log deviations from rolling average volume windows of 7 to 42 days. These shocks proxy abnormal liquidity shifts and capture investor reactions to latent news or market events. Larger positive shocks often reflect institutional trade or market catalysts, while negative values may indicate fading interest or capital flight.

2.1.5 Volatility and Event-Based Feature Validation

We design a set of event-based features that flag extreme movements in weekly price, volatility, and trading volume. These binary signals are engineered to capture episodes of market stress or speculative burst conditions that may trigger short-term reversals or momentum breakdowns.

The input to this module is the weekly-resampled Stage 2 cryptocurrency dataset, containing rolling statistics for return, volume, and volatility. The output is a multi-layered event analysis chart (Figure \ref{fig:event_analysis_btc}), and sample of event impact summary.
(Both graph and table showed in Appendix)
The table suggests that large negative return spikes (jump_down) are frequently followed by positive

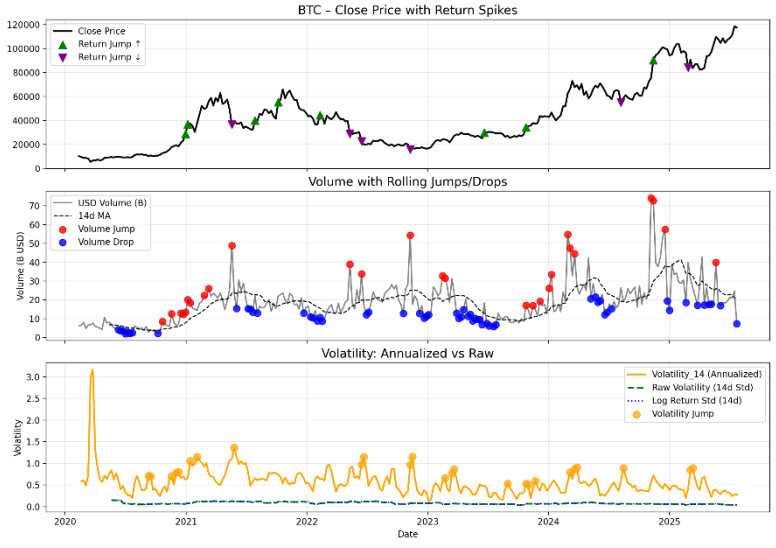


Table 1: BTC Event Impact Summary with Next-Day Returns

Date	Close	Return	Next Rtn	Vol Jump	Vol Drop	Jump ↑	Jump ↓	Volatility Jump
2025-02-26	84197.43	-0.1292	0.0769	False	False	False	True	False
2025-03-05	90671.38	0.0769	-0.0769	False	False	False	False	True
2025-03-12	83701.90	-0.0769	0.0379	False	False	False	False	True
2025-03-26	86939.86	0.0008	-0.0508	False	True	False	False	False
2025-04-16	84057.99	0.0176	0.1150	False	True	False	False	False
2025-04-30	94210.99	0.0052	0.0304	False	True	False	False	False
2025-05-07	97073.38	0.0304	0.0665	False	True	False	False	False
2025-05-21	109686.52	0.0595	-0.0170	True	False	False	False	False
2025-06-04	104729.85	-0.0287	0.0375	False	True	False	False	False
2025-07-23	117574.83	-0.0093	–	False	True	False	False	False

next-day returns (e.g., the first row 2025-02-26, +7.7% after a −12.9% drop), supporting the hypothesis of short-term reversal.

Similarly, volatility spikes often occur near local extrema but tend to normalize within one to two days, implying temporary panic or profit-taking behavior. Volume jumps, on the other hand, show more mixed effects and may require interaction with other features (e.g., momentum) for directional interpretation.

These findings justify the volatility, volume are two important indicators, and the system can include jump_down, volatility_jump, and volume_jump as binary features or filters in Stage 3 portfolio construction, particularly for contrarian strategies.

2.1.6 Data cleaning for Feature Engineer (Structured Data)

- **Handling Missing Data and Delisted Cryptocurrencies:** Any rows with usd_volume <= 0 are converted to NaN before processing to prevent invalid calculations (e.g. log(0)).
- **Delisted Tokens:** Any return values below −100% are assumed to result from delistings or data errors and are removed using: df = df[df["return"] > -1.0].
- **Stablecoins and Wrapped Tokens:** Excluded by applying whitelist filtering. Symbols such as USDT, USDC, WBTC are explicitly dropped.
- **Outlier Detection and Removal:** we capture log returns at a maximum value of 2.

2.2 Unstructured Data Feature Engineering

There are four stages for unstructured data (text) in feature engineering. This process uses the cleaned raw dataset produced in Station 1. Stage 1 involves pulling news within a selected time range; this is completed by our system in Station 1. Stage 2 focuses on text cleaning, including stop word removal and lemmatization. Our system also counts common words and generates a word cloud to extract market sentiment terms. The usage of Stage 2 is to tailor the text dataset into a sentiment grading system, resulting in a VADER-ready dataset. Stage 3 is to run the VADER sentiment and add the VADER score into the dataset. Stage 4 output the word-clouds, confusion matrix. Compound matrix.

2.2.1 Efficient Sentiment Tagging using VADER and FinBERT

Problem: the traditional VADER scoring system is not capable for financial and crypto market words, such as mooning, Diamond hand, Bullish.

Solution: Multi-scoring system VADER and a financial sentiment scoring package FinBERT, filter the financial words that have 0 score (neutral) in VADER, let FinBERT score those words, and combine the score into the VADER compound score.

$$Overall\ Compound\ Score = a * VADER\ compound\ score + (1 - a) * FinBERT\ compound\ score$$

Result: In the total of a = 1.0, 0.7, 0.5, 0.3. And a Max_weighted financial scoring dictionary. **Alpha = 0.5** shows the best performance, which achieved the **best balance** between precision and recall.

Scoring Method	Accuracy	F1 (Macro Avg)	F1 (Weighted Avg)	F1 – Class 1 (Positive)
VADER only (α=1.0)	0.71	0.64	0.68	0.80
VADER 0.7 + FinBERT 0.3	0.72	0.65	0.68	0.81
VADER 0.5 + FinBERT 0.5	**0.75**	**0.68**	**0.72**	**0.83**
VADER 0.3 + FinBERT 0.7	0.74	0.70	0.72	0.82
Max-weighted Hybrid	0.72	0.65	0.68	0.80

Above chart is the tested data set result <week8/stage_1_news_raw.csv.gzip> Alpha = 0.5 is applied on our station 1 news data. The result shows the same trend as the tested set. Code is in Appendix Part A Station 2.

Scoring Method	**Accuracy**	**F1 (Macro Avg)**	**F1 (Weighted Avg)**	**F1 – Class 1 (Positive)**
VADER only (α = 1.0)	0.69	0.65	0.66	0.77
VADER 0.5 + FinBERT 0.5	**0.72**	**0.68**	**0.70**	**0.79**

2.2.2 Event-Driven Token Classification via Granger Causality

According to Balcilar, Sertoglu, and Agan (2022), Granger causality tests revealed a statistically significant causal relationship between news-based COVID-19 sentiment and both the mean and variance of agricultural commodity prices, especially in the extreme quantiles. Inspired by their approach, our system extends the application of Granger causality analysis to the cryptocurrency market, investigating whether news-derived narrative sentiment scores Granger-cause token-level returns.

However, unlike their study, our unstructured dataset presents a data granularity mismatch: while token price data is available at a daily frequency, the volume of narrative-scored news at the daily level is sparse, especially after applying keyword filters and sentiment scoring.

To address this, we adopt a hybrid resampling strategy:

- Narrative sentiment scores are aggregated weekly using the maximum value per week to retain strong signal moments.
- Token returns, in contrast, are computed daily, preserving finer price fluctuation details.
- A complete token × date panel is then constructed by forward filling the weekly scores across daily rows.

Once aligned, we apply Granger causality tests (1-lag) between each narrative score and daily returns for each token. Tokens are then classified into narrative-sensitive groups based on which narrative signals statistically Granger-cause their returns ($p < 0.05$). This enables us to identify clusters such as: ETF-sensitive tokens, Whale-sensitive tokens, Lawsuit- or airdrop-driven assets. This part data shows in the Appendix Part A station 2.

Reference list

Balcilar, M, Sertoglu, K & Agan, B 2022, The COVID-19 effects on agricultural commodity markets, Agrekon, vol. 61, no. 3, pp. 239–265.

Muguto, HT, Muguto, L, Bhayat, A, Ncalane, H, Jack, KJ, Abdullah, S, Nkosi, TS & Muzindutsi, P-F 2022, The impact of investor sentiment on sectoral returns and volatility: Evidence from the Johannesburg stock exchange', Cogent economics & finance, vol. 10, no. 1, pp. 1–24.

Victor Alexandre Padilha, Magnani, V, Rafael Confetti Gatsios, Fabiano Guasti Lima & Rafael Moreira Antonio 2024, Trend and Momentum Technical Indicators for Investing in Market Indices', EkBis Jurnal Ekonomi dan Bisnis, vol. 8, no. 1, pp. 74–86.

Appendix Part A

All path are from the source root <5545Crypto_project> + Output of

Station 1:

1. Stage 1 Structured data

Data processing: <stage1_data/crypto_pipeline.py>
Output of <stage1_data/data_stage1_2_result/results/clean_data/stage_1_crypto_data.csv>

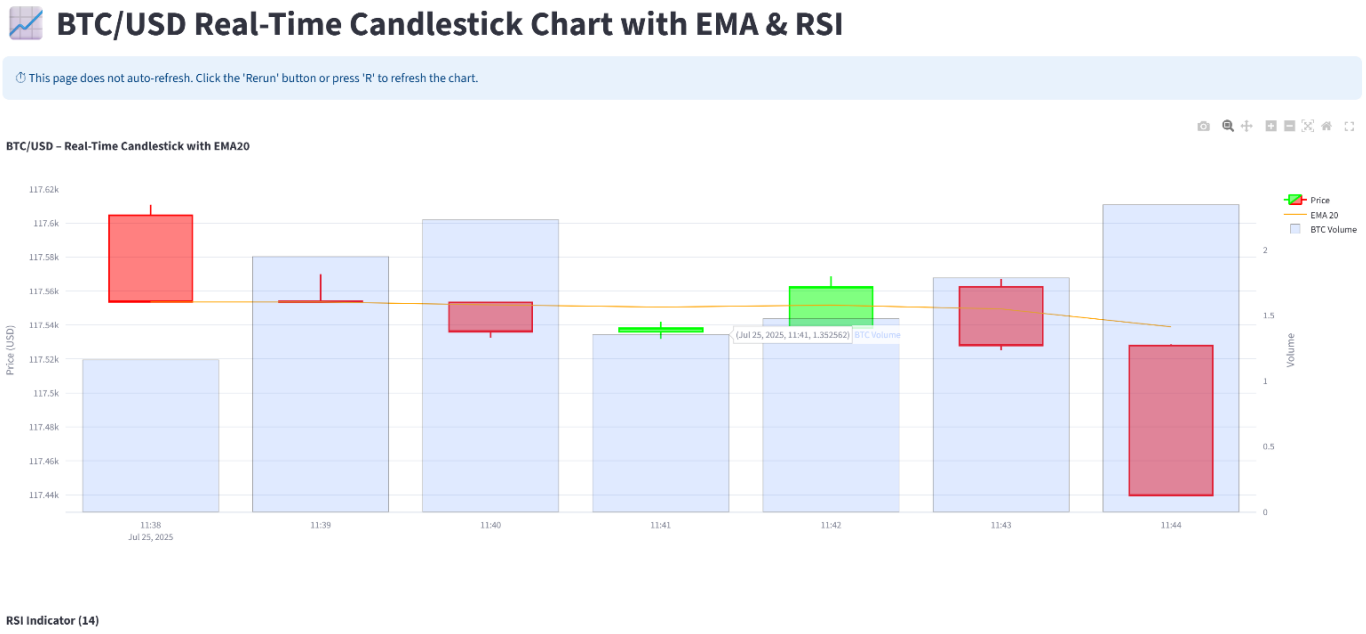
date	id	published_on	title	body	keywords	lang	positive
7/20/2025 14:00	48600435	1753020000	Block: Crypto Mi	Summary Block	XYZ	EN	0
7/20/2025 14:00	48601444	1753020000	Chart of the Wee	Wall Street has	Markets | mar	EN	0
7/20/2025 13:57	48600360	1753019855	Euro debt deals	Emerging-mark	Economy | EU	EN	1
7/20/2025 13:56	48600272	1753019803	XRP price predic	XRP is experien	Cryptocurren	EN	1
7/20/2025 13:54	48600147	1753019650	Bitcoin Price An	Bitcoin (BTC) h	Breaking Nev	EN	0
7/20/2025 13:51	48608392	1753019483	Best Crypto to B	The crypto mar	More News	EN	1
7/20/2025 13:44	48599729	1753019098	Hoskinson prom	Charles Hoskin	Cardano | AD/	EN	0

Real time data catch
Data processing: < stage1_data/realtime_ws_ohlcv.py> and < stage1_data/app_plotly_streamlit.py>
Output of <stage1_data/data/realtime_ohlcv.csv>

Table 1: Sample Real-Time BTC/USD 1-Min OHLCV Data

Symbol	Datetime	Open	High	Low	Close	USD Volume	BTC Volume	USD Vol (mil)
BTC	7/20/2025 8:34	118217.9	118219.9	118217.8	118219.9	2411.01	0.020394	0.002411
BTC	7/20/2025 8:35	118219.9	118221.7	118219.5	118221.7	24921.69	0.210805	0.024922
BTC	7/20/2025 8:36	118221.7	118224.6	118218.0	118218.0	17305.74	0.146388	0.017306
BTC	7/20/2025 8:37	118218.0	118222.7	118215.9	118217.8	5936.0	0.050212	0.005936
BTC	7/20/2025 8:38	118217.8	118253.1	118217.8	118253.1	35535.01	0.300500	0.035535
BTC	7/20/2025 8:39	118253.1	118274.1	118253.1	118274.1	75984.94	0.642448	0.075985

To run the streamlit: cmd C:\Users\Andrea\PycharmProjects\FINS5545\5545Crypto_project\stage1_data>
streamlit run app_plotly_streamlit.py



Unstructured data

Output <stage1_data/data_stage1_2_result/results >

date	id	published_on	title	body	keywords	lang	positive
7/20/2025 14:00	48600435	1753020000	Block: Crypto Mi	Summary Block	XYZ	EN	0
7/20/2025 14:00	48601444	1753020000	Chart of the Wee	Wall Street has	Markets | mar	EN	0
7/20/2025 13:57	48600360	1753019855	Euro debt deals	Emerging-mark	Economy | EU	EN	1
7/20/2025 13:56	48600272	1753019803	XRP price predic	XRP is experien	Cryptocurren	EN	1
7/20/2025 13:54	48600147	1753019650	Bitcoin Price An	Bitcoin (BTC) ha	Breaking New	EN	0
7/20/2025 13:51	48608392	1753019483	Best Crypto to B	The crypto mark	More News	EN	1
7/20/2025 13:44	48599729	1753019098	Hoskinson prom	Charles Hoskin	Cardano | AD/	EN	0

Station 2:

Output of <stage2_features/analyze_volume_jump_impact.py>

<stage2_features/volume_jump_analysis>

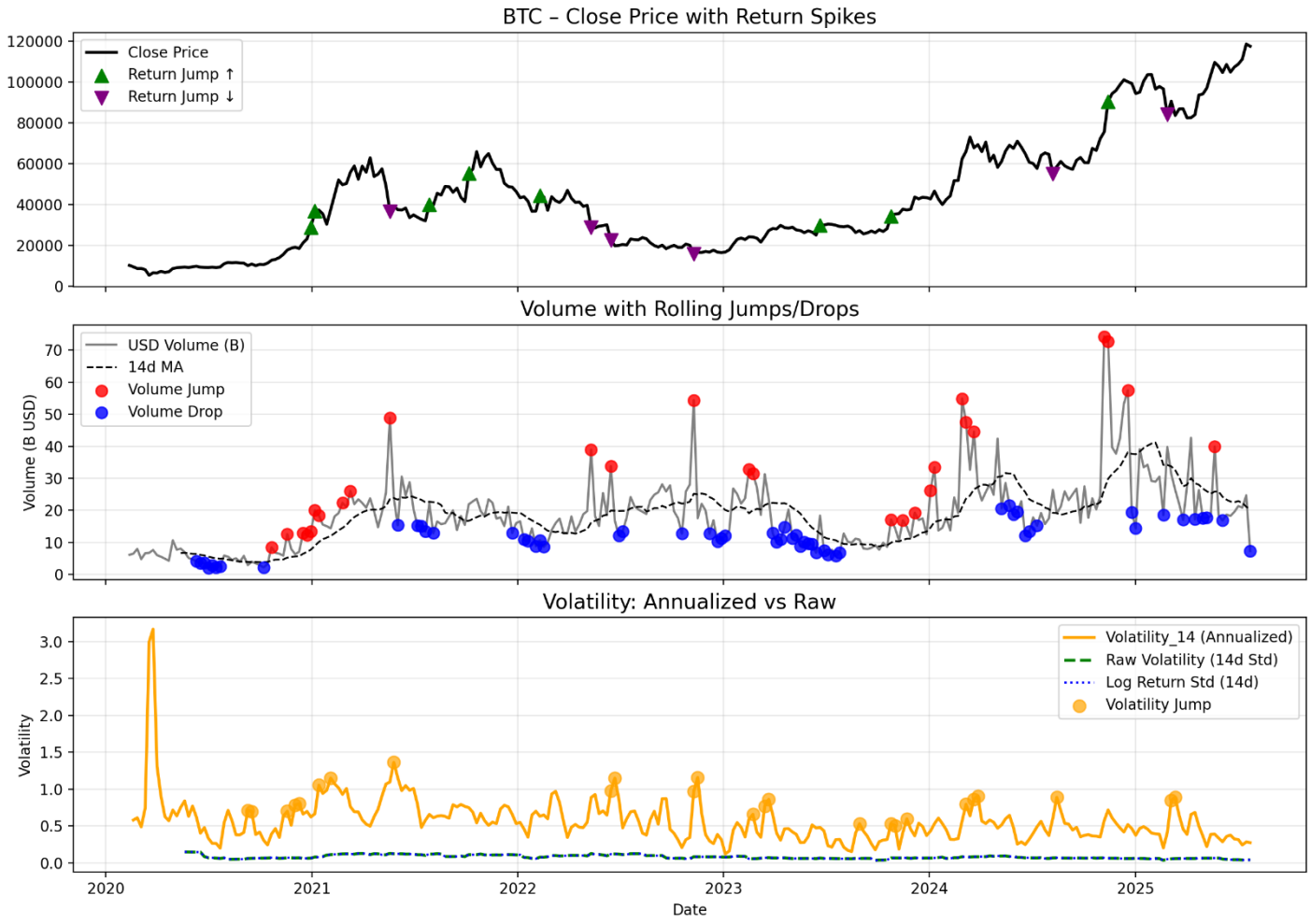
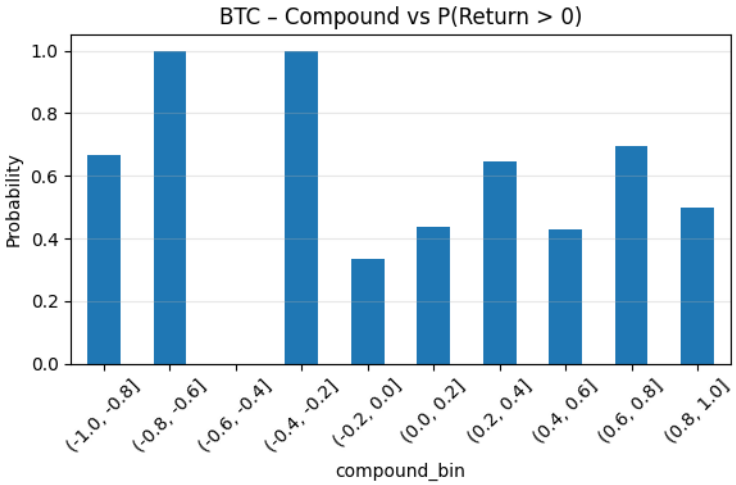


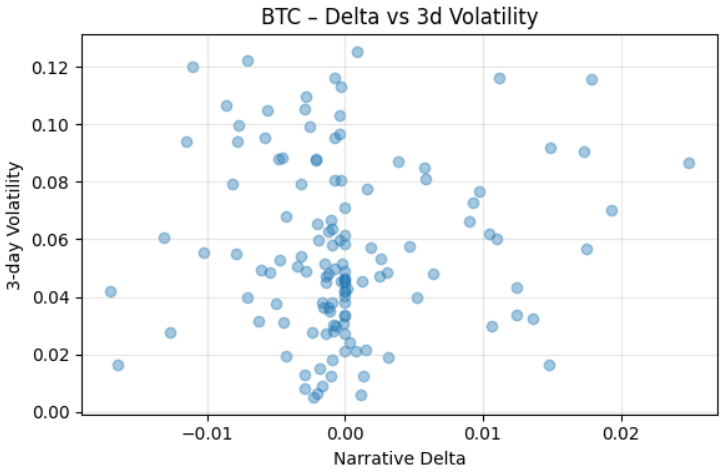
Table 1: BTC Event Impact Summary with Next-Day Returns

Date	Close	Return	Next Rtn	Vol Jump	Vol Drop	Jump ↑	Jump ↓	Volatility Jump
2025-02-26	84197.43	-0.1292	0.0769	False	False	False	True	False
2025-03-05	90671.38	0.0769	-0.0769	False	False	False	False	True
2025-03-12	83701.90	-0.0769	0.0379	False	False	False	False	True
2025-03-26	86939.86	0.0008	-0.0508	False	True	False	False	False
2025-04-16	84057.99	0.0176	0.1150	False	True	False	False	False
2025-04-30	94210.99	0.0052	0.0304	False	True	False	False	False
2025-05-07	97073.38	0.0304	0.0665	False	True	False	False	False
2025-05-21	109686.52	0.0595	-0.0170	True	False	False	False	False
2025-06-04	104729.85	-0.0287	0.0375	False	True	False	False	False
2025-07-23	117574.83	-0.0093	–	False	True	False	False	False

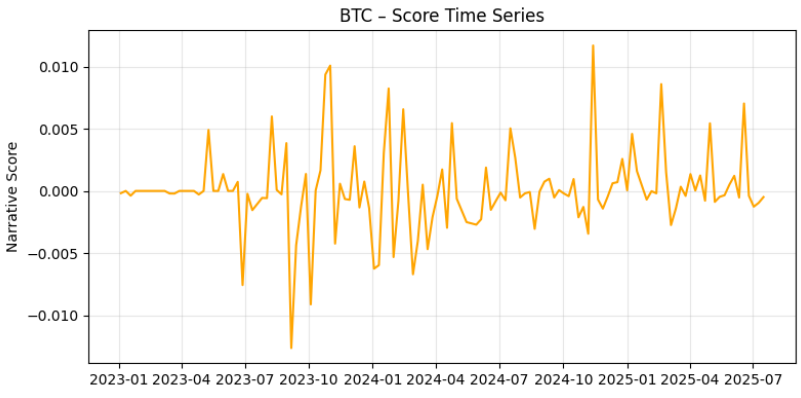
<stage2_features/analyze_volume_jump_impact.py>

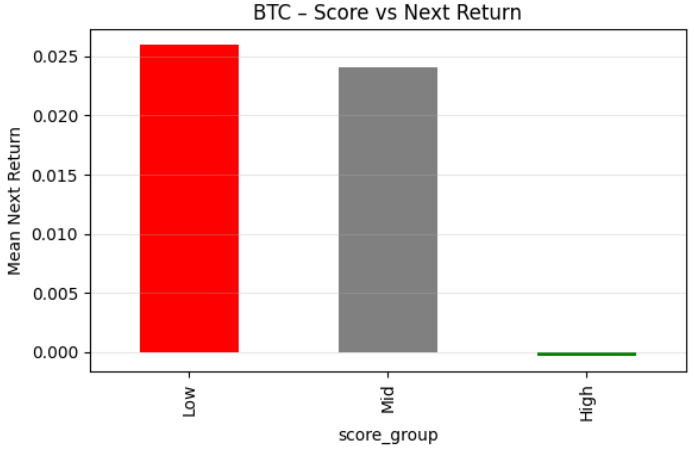
Narrtive analysis output: <stage2_features/narrative_analytics.py>
Example: **BTC & ETF_Event**



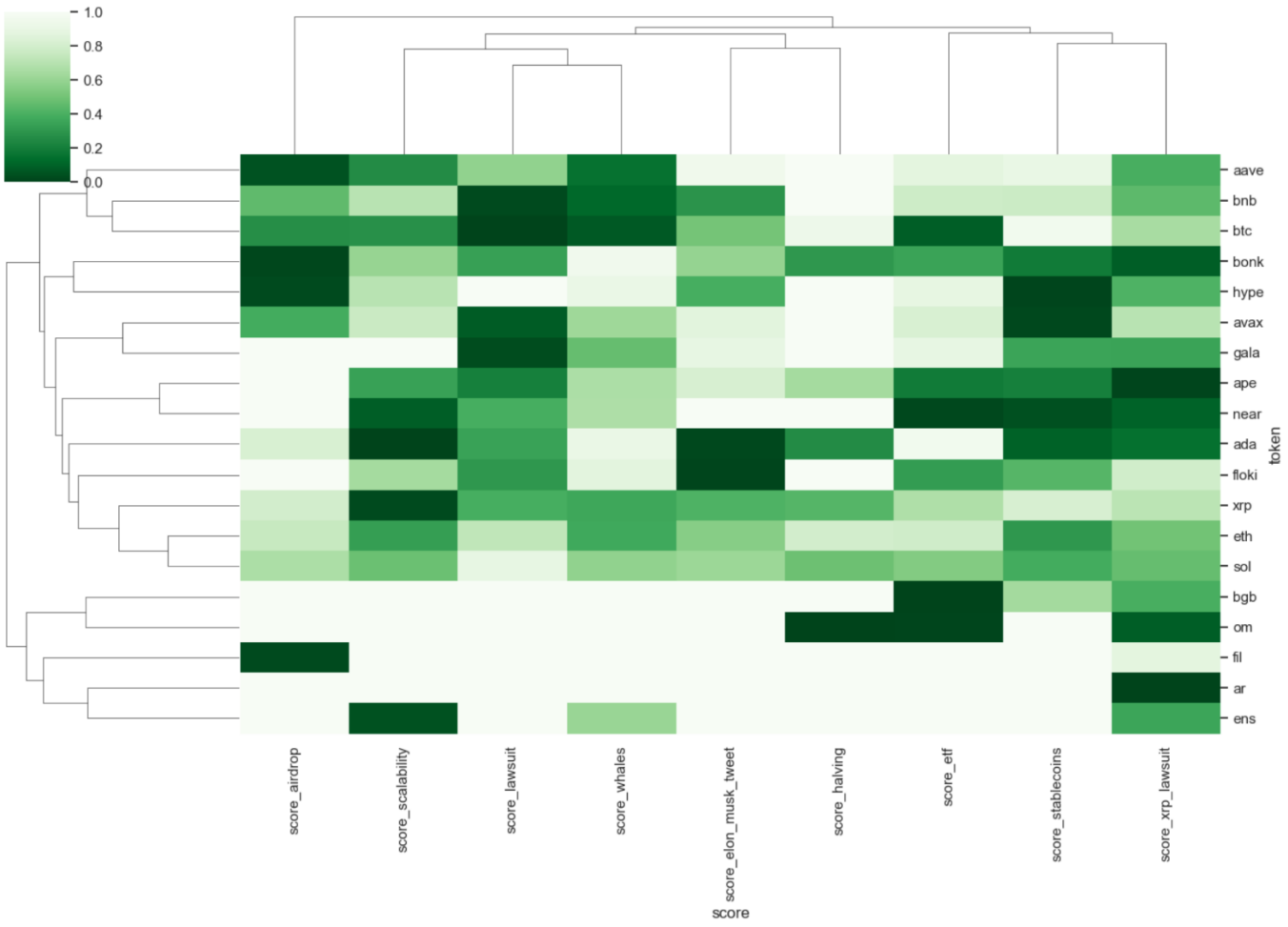


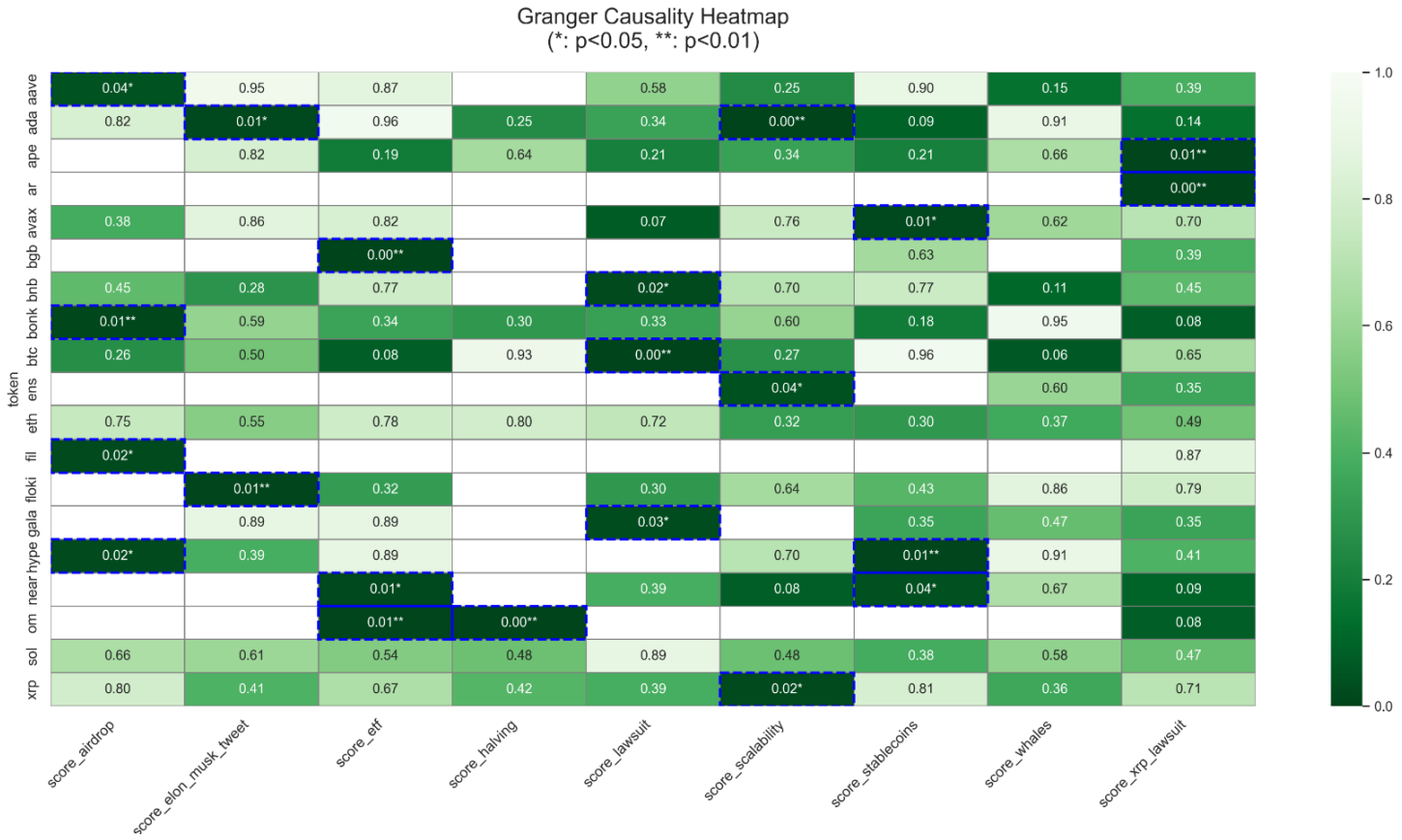
<stage2_features/etf_event/narrative_charts>

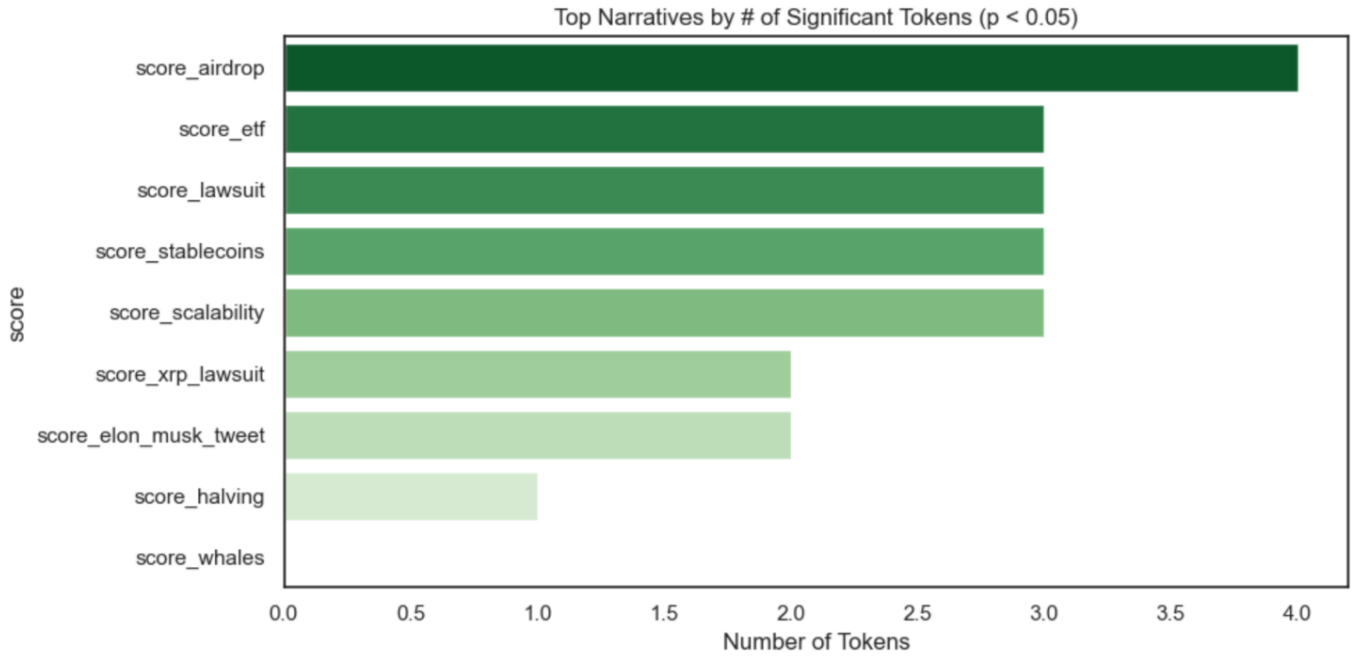




Granger Result <stage2_features/granger_action.py>
Result file: <stage2_features/granger_result>




Granger Causality Heatmap
(*: p<0.05, **: p<0.01)
token
aave 0.04* 0.95 0.87 0.58 0.25 0.90 0.15 0.39
ada 0.82 0.01* 0.96 0.25 0.34 0.00** 0.09 0.91 0.14
ape 0.82 0.19 0.64 0.21 0.34 0.21 0.66 0.01**
ar 0.00**
avax 0.38 0.86 0.82 0.07 0.76 0.01* 0.62 0.70
bgb 0.00** 0.63 0.39
bnb 0.45 0.28 0.77 0.02* 0.70 0.77 0.11 0.45
bonk 0.01** 0.59 0.34 0.30 0.33 0.60 0.18 0.95 0.08
btc 0.26 0.50 0.08 0.93 0.00** 0.27 0.96 0.06 0.65
ens 0.04* 0.60 0.35
eth 0.75 0.55 0.78 0.80 0.72 0.32 0.30 0.37 0.49
fil 0.02* 0.87
floki 0.01** 0.32 0.30 0.64 0.43 0.86 0.79
gala 0.89 0.89 0.03* 0.35 0.47 0.35
hype 0.02* 0.39 0.89 0.70 0.01** 0.91 0.41
near 0.01* 0.39 0.08 0.04* 0.67 0.09
om 0.01** 0.00** 0.08
sol 0.66 0.61 0.54 0.48 0.89 0.48 0.38 0.58 0.47
xrp 0.80 0.41 0.67 0.42 0.39 0.02* 0.81 0.36 0.71
score_airdrop
score_elon_musk_tweet
score_etf
score_halving
score_lawsuit
score_scalability
score_stablecoins
score_whales
score_xrp_lawsuit
1.0
0.8
0.6
0.4
0.2
0.0


Top Narratives by # of Significant Tokens (p < 0.05)
score
score_airdrop
score_etf
score_lawsuit
score_stablecoins
score_scalability
score_xrp_lawsuit
score_elon_musk_tweet
score_halving
score_whales
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Number of Tokens

```
C:\Users\Andrea\PycharmProjects\FINS5545\.venv\Scripts\python.exe C:\Users\Andrea\PycharmProjects\FINS5545\5545Cry
✅ Granger test completed. Results saved to: C:\Users\Andrea\PycharmProjects\FINS5545\5545Crypto_project\stage2_fe
📊 Heatmap saved to: C:\Users\Andrea\PycharmProjects\FINS5545\5545Crypto_project\stage2_features\granger_result\gr
📎 Narrative Cluster Summary:
Theme Cluster 1: includes score_scalability, score_lawsuit, score_whales, score_elon_musk_tweet, score_halving
Theme Cluster 2: includes score_stablecoins, score_xrp_lawsuit
Theme Cluster 3: includes score_etf
Theme Cluster 4: includes score_airdrop

✅ All-token clustering complete. Sample:
   token token_group
0  1inch     Group B
1   aave     Group E
2    ada     Group D
3   aero     Group A
4   aioz     Group B

📊 Token counts per group:
token_group
Group A    77
Group E    26
Group B    16
Group C    10
Group D     8
Group F     1
Name: count, dtype: int64

📊 Token Group Sensitivity Summary (FULL TOKEN SET):
Group A tokens are most sensitive to: score_whales, score_lawsuit, score_xrp_lawsuit
Group B tokens are most sensitive to: score_etf, score_xrp_lawsuit, score_airdrop
Group C tokens are most sensitive to: score_airdrop, score_elon_musk_tweet, score_etf
Group D tokens are most sensitive to: score_elon_musk_tweet, score_scalability, score_stablecoins
Group E tokens are most sensitive to: score_stablecoins, score_whales, score_lawsuit
Group F tokens are most sensitive to: score_halving, score_etf, score_xrp_lawsuit
🔁 Reassigned om from Group F → Group B

Process finished with exit code 0
```

Sentiment: VADER & FinBERT smart score code
Note: the tested VADER & FinBERT result is in week 8 folder

News result: <stage1_data/news_results>
News result after apply VADER & Finbert: <stage1_data/_project_finbert_a05_sentiment_result>

Sentiment pipeline: stage1_data/finbert_a05_sentiment_pipeline.py

Test finbert result is in week 8 folder:
Alpha = 0.3 --→ week8/finbert_a03_week8
Alpha = 0.5 --→ week8/finbert_a05_week8
Alpha = 0.7 --→ week8/finbert_a07_week8
Alpha = 0.1 --→ week8/text_week8
Max weighted method -→ week8/max_weighted_text_week8

```python
# Download VADER lexicon if needed
import nltk
from transformers import pipeline
from nltk.sentiment.vader import SentimentIntensityAnalyzer
import re
import pandas as pd

nltk.download("vader_lexicon")

# Define FinBERT pipeline and cache
_FINBERT = pipeline("sentiment-analysis", model="yiyanghkust/finbert-tone",
tokenizer="yiyanghkust/finbert-tone")
_finbert_cache = {}

# Define finance-related keywords that VADER is insensitive to
FINANCE_KEYWORDS = {
    "bullish", "bearish", "mooning", "rekt", "hodl", "fomo", "fud", "degen",
    "rugpull", "diamondhands", "paperhands", "pump", "dump", "shill",
    "crash", "plummet", "collapse", "drawdown", "correction", "rebound",
    "spike", "whipsaw", "moon", "skyrocket", "surge", "soar",
    "short squeeze", "liquidation", "panic", "fear", "greed",
    "buy the dip", "euphoria", "whale", "weak hands", "strong hands"
}

_VADER = SentimentIntensityAnalyzer()

def _smart_score(txt: str, alpha: float = 0.5) -> pd.Series:
    """
    Hybrid sentiment scoring function combining:
        - VADER (lexicon + rules-based)
        - FinBERT (transformer-based financial context model)

    Adds FinBERT adjustment only if finance-specific slang is detected
    and not covered by VADER lexicon. Uses exponential weighting via alpha.
    Returns: pd.Series with 'neg', 'neu', 'pos', and final 'compound' score.
    """
    vader_scores = _VADER.polarity_scores(txt)
    compound = vader_scores["compound"]

    # Clean text
    clean_text = re.sub(r"[^\w\s]", "", txt.lower())
    tokens = set(clean_text.split())
    present_keywords = tokens & FINANCE_KEYWORDS

    # Check if FinBERT is needed for domain-specific adjustment
    needs_finbert = any(_VADER.lexicon.get(w, 0.0) == 0.0 for w in present_keywords)

    if needs_finbert:
        if txt in _finbert_cache:
            f_score = _finbert_cache[txt]
        else:
            label = _FINBERT(txt[:512])[0]["label"].lower()
            f_score = {"positive": 1.0, "neutral": 0.0, "negative": -1.0}[label]
            _finbert_cache[txt] = f_score

        compound = round(alpha * compound + (1 - alpha) * f_score, 4)

    return pd.Series({
        "neg": vader_scores["neg"],
        "neu": vader_scores["neu"],
        "pos": vader_scores["pos"],
        "compound": compound
    })
eed, [])
}
```

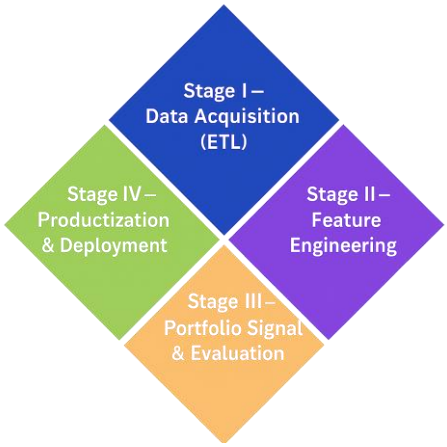
ZENTRYX

Executive Summary

OVERVIEW

Zentryx is a sentiment-enhanced cryptocurrency strategy engine designed to generate optimized portfolio allocation for investors. There are total of seven strategies and categorized as conservative, balanced, and aggressive trading style.

This report focuses on **Stage III (Portfolio Signal & Evaluation)** and **Stage IV (Productization & Deployment)**. This report outlines how Zentryx improves the accuracy of sentiment scores to enhance trading signal reliability. It simulates portfolio allocation across 15 cryptocurrencies, applying ML-models to predict future price trends and evaluate strategy performance. Finally, Zentryx trading strategy engine deployed through Google Gemini:



[Zentryx Demo link]

KEY FINDINGS AND RECOMMENDATIONS

Structured Data

1. Total of six trading strategies based on structured data, achieving high annualized return.
2. All trading strategies were backtested from 2020-08-01 to 2025-07-31, producing performance visualizations with actionable trading signals

Unstructured Data

1. Developed a FinBERT+VADER hybrid sentiment pipeline with 0.73 F1-score and 96% positive recall.
2. Two-year sentiment overview by token category, with weekly and monthly resampling.
3. Dynamic sentiment trading rotates monthly sentiment-selected tokens, delivering a 179% annualized return.

Contents

ZENTRYX

STATION 3:
Model Design and Sentiment Analysis

3.1 Model Selection and Design

To transform engineered structured features into systematic portfolio signals, Zentryx integrates a diversified set of trading strategy archetypes. Each strategy exploits distinct return-generating mechanisms derived from technical indicators, volatility patterns, and sentiment signals. We define six core portfolio strategies, including Maximum Sharpe Ratio (MSR), Minimum Variance (MV), MVRA (Min-Variance + Risk Aversion), Equal Weight (EW), Risk Parity (RP), Sentiment-Tilted, Momentum-Weighted, and Volatility Targeting Portfolio. Each corresponding to different investor beliefs about market dynamics from defensive to speculative portfolios.

Client's portfolio:

"BTC", "ETH", "SOL", "DOGE", "MATIC",
"KAS", "LDO", "UNI", "AAVE", "MKR",
"BNB", "TAO", "ADA", "PEPE", "BONK"

3.1.1 Maximum Sharpe Ratio (MSR)

In cryptocurrency trading, a significant portion of participants are motivated by high-risk, high-reward speculation. Many cryptocurrencies exhibit highly positively skewed return distributions — a "get-rich-quick" effect, which may attract investors with a gambling propensity. MSR strategy uses these special market features to allocate portfolio with maximized Sharpe ratio, aiming to capitalize on sharp upward price movements while bearing considerable risk.

$$\max_{w} \frac{w^\top \mu - r_f}{\sqrt{w^\top \Sigma w}} \quad \text{subject to} \sum_{i=1}^{N} w_i = 1,\ w_i \geq 0 \tag{1}$$

Code implements with two different methods: NumPy and CVXY

```python
# === Strategy 1: MSRP (Max Sharpe Ratio Portfolio) numpy ===
inv_cov = np.linalg.pinv(cov)
w_msr = inv_cov @ mu
w_msr /= w_msr.sum()
save_weights(pd.Series(w_msr, index=symbols), "msrp")
```

```python
# === Strategy1 MSRP CVXY ===
w_msr = cp.Variable(n)
t = cp.Variable()
try:
    L = np.linalg.cholesky(cov.values)
except:
    raise ValueError("Covariance matrix not positive-definite.")

objective = cp.Maximize(mu.values @ w_msr)
constraints = [cp.sum(w_msr) == 1, w_msr >= 0, w_msr <= 0.2, cp.norm(L @ w_msr, 2) <= t, t == 1]
prob = cp.Problem(objective, constraints)
prob.solve()
if w_msr.value is not None:
    save_weights(pd.Series(w_msr.value, index=symbols), "msrp"
```

Table 1: Performance Comparison of MSRP Strategy Implementations

Metric	MSRP (cvxpy)	MSRP (NumPy)
Annual Return	4.4186	3.3526
Annual Volatility	0.7697	0.9724
Sharpe Ratio	5.7415	3.3448
Max Drawdown	-0.8348	-0.7067
CVaR (95%)	-0.2213	-0.1942

Inspired by Qu and Zhang (2023), I implemented MSPR strategy by adding more constraints, such as individual asset weight caps and L2-norm-based risk control. In particular, the upper bound constraint w<= 0.2 was imposed to prevent excessive concentration in a single asset. Under the latest five-year backtest, the constrained cvxpy-based MSRP outperformed the NumPy approximation, achieving an annualized return of 441.86%, Sharpe ratio of 0.76.

3.1.2 Minimum Variance (MV)

The second strategy is minimum variance portfolio, which is suitable for risk-averse investors seeking portfolio stability. In the study by Qu and Zhang (2023), the GMVP strategy selected the most stable assets AMZN, BRK-B, and WMT during the volatile U.S. stock market period in 2021. Therefore, Zentryx is using MVP to advise low-risk investment suggestions.

$$\min_{w} w^{\top}\Sigma w \quad \text{subject to} \quad \sum_{i=1}^{N} w_i = 1, \quad w_i \geq 0 \tag{2}$$

Code Showcase:

```
# === Strategy 2: GMVP (Global Minimum Variance Portfolio) ===
w_gmvp = inv_cov @ np.ones(n)
w_gmvp /= w_gmvp.sum()
save_weights(pd.Series(w_gmvp, index=symbols), "gmvp")
```

3.1.3 Equal Weighted Portfolio (EW) & Risk Parity (Inverse Volatility Approximation)

The third and the fourth strategy is Equal weighted and risk parity. Both strategies do not rely on expected return, investors will use those strategies when there are no strong assumptions about expected return. Equal Weighted strategy allocates capital uniformly across assets regardless of their risk, the Risk Parity strategy allocates risk equally.

$$w_i = \frac{1}{N}, \quad \forall i \in \{1, 2, ..., N\} \tag{3}$$

$$w_i = \frac{1/\sigma_i}{\sum_{j=1}^{N} 1/\sigma_j} \text{where } \sigma_i \text{ is the standard deviation of asset } i \tag{4}$$

```
# === Strategy 4: Equal Weighted ===
w_eq = pd.Series(1 / n, index=symbols)
save_weights(w_eq, "equal_weighted")

# === Strategy 6: Risk Parity Approximation ===
inv_vol = 1 / asset_vol
w_rp = inv_vol / inv_vol.sum()
save_weights(pd.Series(w_rp, index=symbols), "risk_parity_approx")
```

Table 2: Performance Comparison: Equal Weighted Variants and Risk Parity

Metric	Equal Weighted	Risk Parity	Equal Weighted Top-7 Momentum
Annual Return	1.2945	1.1122	1.7977
Annual Volatility	0.7076	0.6837	0.8077
Sharpe Ratio	1.8294	1.6268	2.2258
Max Drawdown	-0.6589	-0.6922	-0.6192
CVaR (95%)	-0.1752	-0.1797	-0.1698

Continuing implementation from the traditional method, Zentryx added a momentum factor into the equal-weight strategy. The result shows a significant improvement in **annual return**, increasing from **1.2945** to **1.7977**. The top 7 tokens were selected monthly based on their annual momentum scores, calculated as: `momentum_scores = (returns.add(1).prod() ** (1 / len(returns))) - 1` However, the **Sharpe Ratio** has also increased slightly, and the **Max Drawdown** shows better downsides protection, improving from **-0.6589 to -0.6192**. The **CVaR (95%)** has also improved, from -0.1752 to -0.2009, which highlights enhanced resilience during extreme market conditions. Therefore, Zentryx

decided to keep strategy Equal Weighted Top 7 Momentum.

3.1.4 Maximum Diversification Ratio Portfolio (MDRP)

To enhance portfolio stability without compromising returns, Zentryx incorporates the **Maximum Diversification Ratio strategy**. This approach selects weights that maximize diversification benefits by **minimizing correlations among assets**.

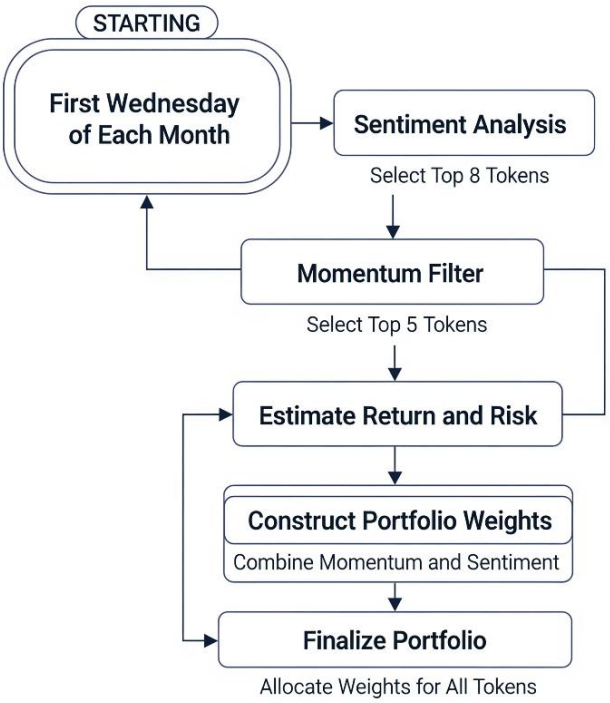
$$w = \frac{\Sigma_{\text{corr}}^{-1}\mathbf{1}}{\mathbf{1}^{\top}\Sigma_{\text{corr}}^{-1}\mathbf{1}} \tag{5}$$

Table 3: Performance Comparison of Portfolio Strategies (structural)

Strategy	Annual Return	Annual Volatility	Sharpe Ratio	Max Drawdown	CVaR (95%)
MSRP	3.2526	0.9724	3.3448	-0.7067	-0.1942
GMVP	0.5542	0.5360	1.0338	-0.7657	-0.1572
Risk Parity	1.1122	0.6837	1.6268	-0.6922	-0.1797
Equal Weighted	1.2945	0.7076	1.8294	-0.6589	-0.1752
Equal Weighted Top-7 MOM	1.7977	0.8077	2.2258	-0.6192	-0.1698
Max Diversification	1.1122	0.6837	1.6268	-0.6922	-0.1797
BTC	0.6028	0.6094	0.9892	-0.7590	-0.1707

As shown in table 3, The Maximum Diversification Ratio strategy stands out for its strong downside protection. It achieved a low CVaR of -0.2080 and a moderate max drawdown of -0.8789, both better than most benchmark strategies. This makes it ideal for investors who seek steady returns with reduced tail risk.

3.1.5 Dynamic Sentiment-Tilted Strategy with Momentum Filter



This strategy dynamically adjusts portfolio allocations on a **monthly basis**, combining sentimental signals from financial news with recent **price momentum** to identify outperforming tokens. The client's portfolio allocation dynamically adjusts every month.

Selection process:

1. **Sentiment Extraction**: At the start of each month (aligned to the **first Wednesday**), we extract the latest available sentiment scores for all tokens. These scores are standardized into **Z-scores** to account for scale differences and identify the **top 8 tokens** with the **strongest positive sentiment**. → Result: Select the **top 8 tokens** with the highest positive sentiment
2. **Momentum Filtering:** Based on the top 8 sentimental scoring tokens selected in the first stage, we calculate **30-day cumulative returns** for the tokens. →Result: Select the **top 5 tokens** with the strongest momentum.

3. **Portfolio Weight Construction:**

$$w_i = (1-\alpha)\cdot\frac{m_i}{\sum_j m_j} + \alpha\cdot\frac{s_i}{\sum_j s_j}$$

$$\text{where} \quad m_i = \text{Momentum}_{30\text{d}}(i), \quad s_i = \text{Z-Score Sentiment}(i)$$

Table 5: Performance of Sentiment-Tilted Portfolios with Different α Values

Strategy	Annual Return	Annual Volatility	Sharpe Ratio	Max Drawdown	CVaR (95%)
Sentiment-Tilted ($\alpha = 0.3$)	1.9107	0.8561	2.2319	-0.5954	-0.1438
Sentiment-Tilted ($\alpha = 0.5$)	1.7833	0.8367	2.1314	-0.5997	-0.1537
Sentiment-Tilted ($\alpha = 0.7$)	1.6437	0.8270	1.9876	-0.6046	-0.1650

As shown in Equation, this soft-tilt mechanism allows dynamic adjustment of allocations based on both monthly price trends and news-driven sentiment signals. In the backtesting stage, we tested $a = 0.3, 0.5, 0.7$.

When $a = 0.3$, it shows the best performance, yielding the highest annual return (1.91) and Sharpe ratio (2.2319). It also maintains favorable downside protection, with a smaller max drawdown and more conservative CvaR.

3.2 Sentiment Analysis Implementation & Strategy

3.2.1 Efficient Sentiment Tagging using VADER and FinBERT

As discussed previously in part A, there are limitations of the traditional VADER sentiment analysis in handling domain-specific financial and cryptocurrency terms (e.g., *mooning*, *diamond hand*, *bullish*), we implemented a **hybrid sentiment scoring system** that combines VADER with a financial-domain language model, **FinBERT**.

Specifically, the method filters words that receive a neutral score (0) from VADER and re-evaluates them using FinBERT. The final compound sentiment score is computed as a weighted combination:

$$\text{Compound Score} = \alpha \cdot \text{VADER}_{\text{compound}} + (1-\alpha)\cdot \text{FinBERT}_{\text{compound}}$$

The results demonstrated that α = 0.5 achieves the best trade-off between precision and recall, yielding the highest weighted F1 score and accuracy:

Scoring Method	Accuracy	F1 (Macro Avg)	F1 (Weighted Avg)	F1 – Class 1 (Positive)
VADER only (α = 1.0)	0.69	0.65	0.66	0.77
VADER 0.5 + FinBERT 0.5	**0.72**	**0.68**	**0.70**	**0.79**

3.2.2 Sentiment Data Implementation

Zentryx is using sentiment analysis based on the VADER 0.5 + FinBERT 0.5 to work on the sentiment aligned strategy. To incorporate unstructured news data into our investment strategy, Zentryx automatically extracts referenced cryptocurrencies from news articles and generates daily, weekly, and monthly sentiment time series for each token. This process is achieved by constructing a crypto keyword mapping dictionary and matching each article's text. We also resembled monthly (**first Wednesday**) frequencies to align with the price data in our backtesting framework. The output is the wide format, with the compound scores in the columns, each column represents different tokens. The example data is below:

Table 4: Example Monthly Sentiment Scores by Token

Date	BTC	ETH	SOL	DOGE	MATIC	KAS	UNI	ADA
2023-01-04	0.2809	0.3201	0.4888	0.6011	0.2423	-0.1444	0.5023	0.4270
2023-02-01	0.4649	0.5582	0.4862	0.6117	0.5329	0.9995	0.5374	0.6609
2023-03-01	0.2500	0.4281	0.1988	0.4847	0.5568	0.8964	0.4796	0.4059
2023-04-05	0.4005	0.3518	0.3953	0.3562	0.4355	-0.4871	0.5143	0.4334
2023-05-03	0.3804	0.4885	0.5464	0.3667	0.5963	0.9898	0.5494	0.3388

3.3 Technical Constraints and Limitations

3.3.1 Sentiment Strategy Limitations

Although the strategy provides dynamic monthly allocations, it is still subject to **sentiment lag**. If a token experienced strong sentiment and momentum mid-month, our model may select it for investment in the following month, potentially after the peak—leading to poor entry timing in a volatile market. To improve this, future versions could integrate **real-time sentiment checks**. If sentiment remains overheated and indicators like RSI suggest overbought conditions, the strategy can adjust by **reducing or delaying the allocation**.

Another limitation is that the bull markets are often counter sentiment driven and inherently difficult to predict. According to Tiwari et al. (2022), it indicates that consumer sentiment can predict sectoral stock returns, particularly under normal market conditions. However, the prediction through sentiment weakens in extreme bull or bear markets. In the future integration of sentiment, we can add the social media sentiment from platforms like Reddit, Twitter (X), Facebook, and even Telegram or TikTok. Tweets from celebrities, politicians, and business leaders are often important factors influencing sentiment scores. With the expansion of sentiment data sources, continuous development of technical infrastructure is necessary to strengthen the effectiveness of sentiment-driven trading strategies.

3.3.2 Structural Strategy Limitations

Extreme market conditions: Just like sentiment part, Zentryx's structural strategies also lack the ability to foresee extreme market conditions, such as bull runs and Black Swan Event. In earlier testing, volatility and event features did provide meaningful historical trading signals, but these were built with the benefit of having past data in hand. In the next stage of development, Zentryx will strengthen its price forecasting capabilities, combining them with live market sentiment, and use these forecasts to test how well existing strategies respond to market signals before they happen.

Low-Liquidity and momentum environments: For thinly traded altcoins, sentiment shocks can be amplified by illiquidity, leading to unstable allocations and higher slippage than backtests imply. Meanwhile, the low liquidity condition would bring low momentum, many of my strategies have the process to rank the holding coins by price momentum. In future iterations, Zentryx will upgrade its momentum ranking process to benchmark not only the client's existing holdings but also the broader market. This enhanced ranking will allow the system to recommend refreshed investment portfolios that are informed by the strongest market-wide momentum signals, rather than being limited to the current set of holdings.

Narrative Saturation: When a narrative (e.g., ETF approval) dominates news flow for weeks, diminishing marginal predictive power may cause the model to overweight affected tokens without incremental return benefit. In my narrative granger test, I keep finding the reason why XRP and event_xrp_lawsuit & event_lawsuit have no correlation, I think the reason is that the lawsuit has too many new flows, which would influence the final result.

Time Horizon limitations with rebalancing lag: Although a weekly rebalance has delivered strong performance in five-year historical tests by reducing short-term noise, the cryptocurrency market is highly sensitive and can experience sharp rallies and reversals within a matter of hours. This lag makes it easy to miss short-term trading opportunities that unfold and fade before the next rebalance. When trade decisions are made only in the following week, there is a heightened risk of entering positions after peak sentiment has already passed and missing optimal exit points.

ZENTRYX

STATION 4
Model Implementation

4. Implementation Steps Explanation & Canvas Showcase

Google Gemini result: [Zentryx Demo link]

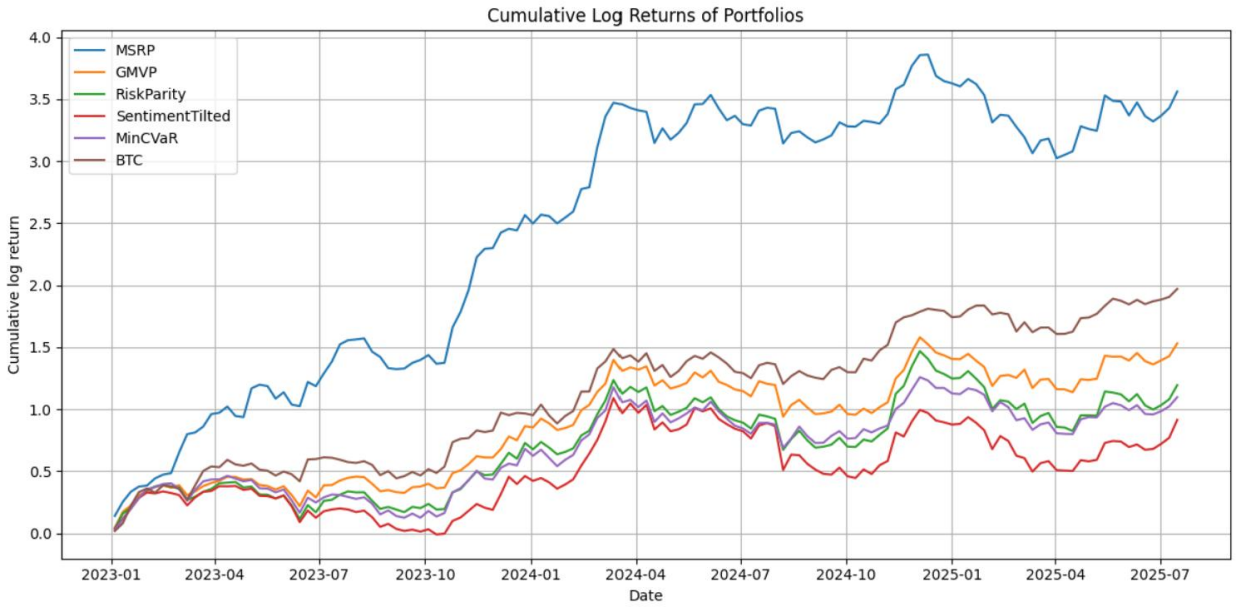
4.1 Cryptocurrency portfolio optimization model training and validation

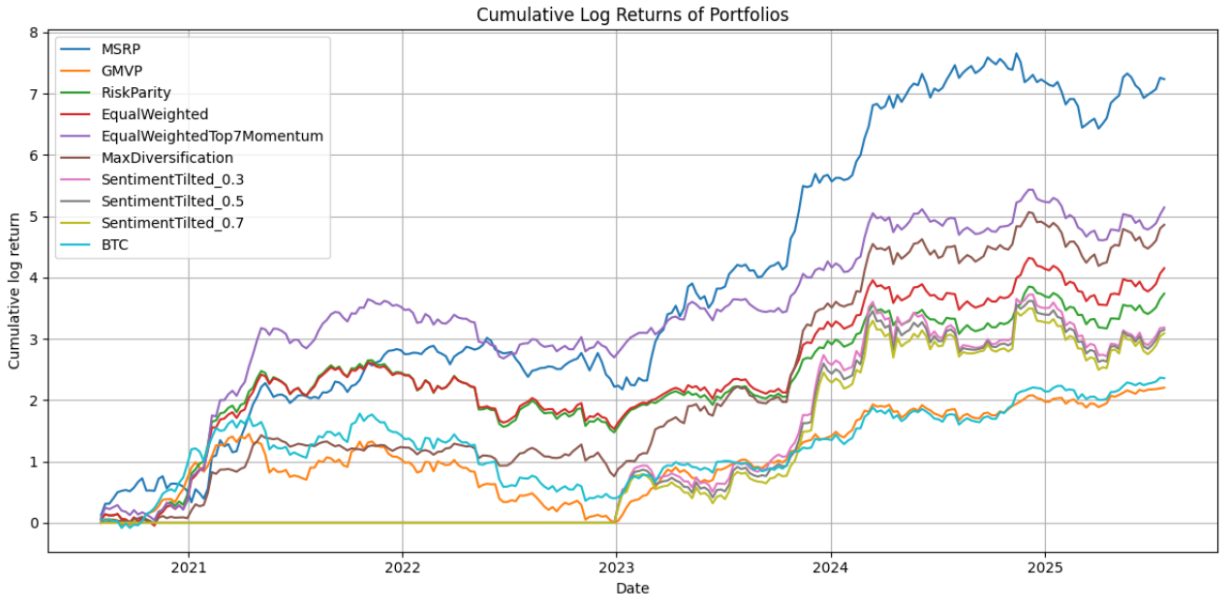
Zentryx includes a total of seven portfolio strategies, all backtested over a five-year period from 2020-08-01 to 2025-07-31. The workflow runs sequentially as follows:

- cvxy_strategy.py (base strategy construction; outputs monthly allocation weights)
- backtest_cvxy_strategy.py (performs backtesting; outputs portfolio_returns_combined.csv)
- portfolio_result.py (visualizes strategy performance; generates strategy_metrics.csv overview for Station 3)
- client_token_result.py (plots individual strategy results across tokens)

Initially, I implemented constraints using cvxpy, particularly in the MSRP strategy. However, as the system became more complex, cvxpy introduced frequent computational errors and performance bottlenecks. To improve stability and scalability, I transitioned the remaining strategies to a NumPy-based implementation.

Although the cvxpy version of MSRP outperformed BTC in terms of cumulative log return, the overall improvement across strategies was limited. The switch to NumPy proved worthwhile—it simplified debugging, accelerated development, and still delivered competitive performance, especially in the dynamic sentiment strategy. The first image shows results from the cvxpy implementation; the second reflects the refined NumPy-based version.

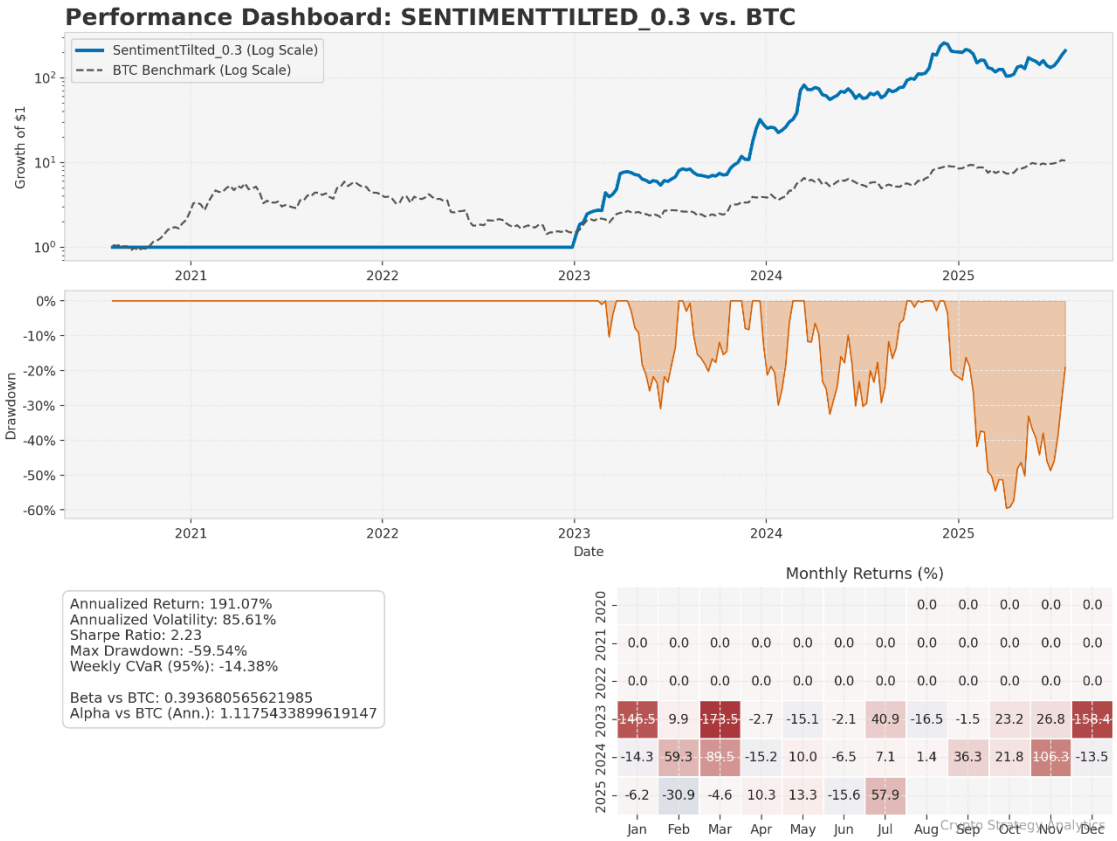




4.2 VADER sentiment analysis implementation and testing

I implemented a hybrid sentiment scoring method (alpha = 0.5) in finbert_a05_sentiment_pipeline.py under the Stage 1 folder. The model combines VADER and FinBERT to enhance financial text sentiment accuracy. I tested multiple alpha values (0.3, 0.5, 0.7, 1.0) and found 0.5 achieved the best F1 performance. The final output is a token-level sentiment time series, from which I extract both monthly average scores and token-specific sentiment indicators—key inputs for my dynamic sentiment-tilted portfolio strategy.

1. Sentiment index construction and validation & Integration of sentiment signals with portfolio optimization

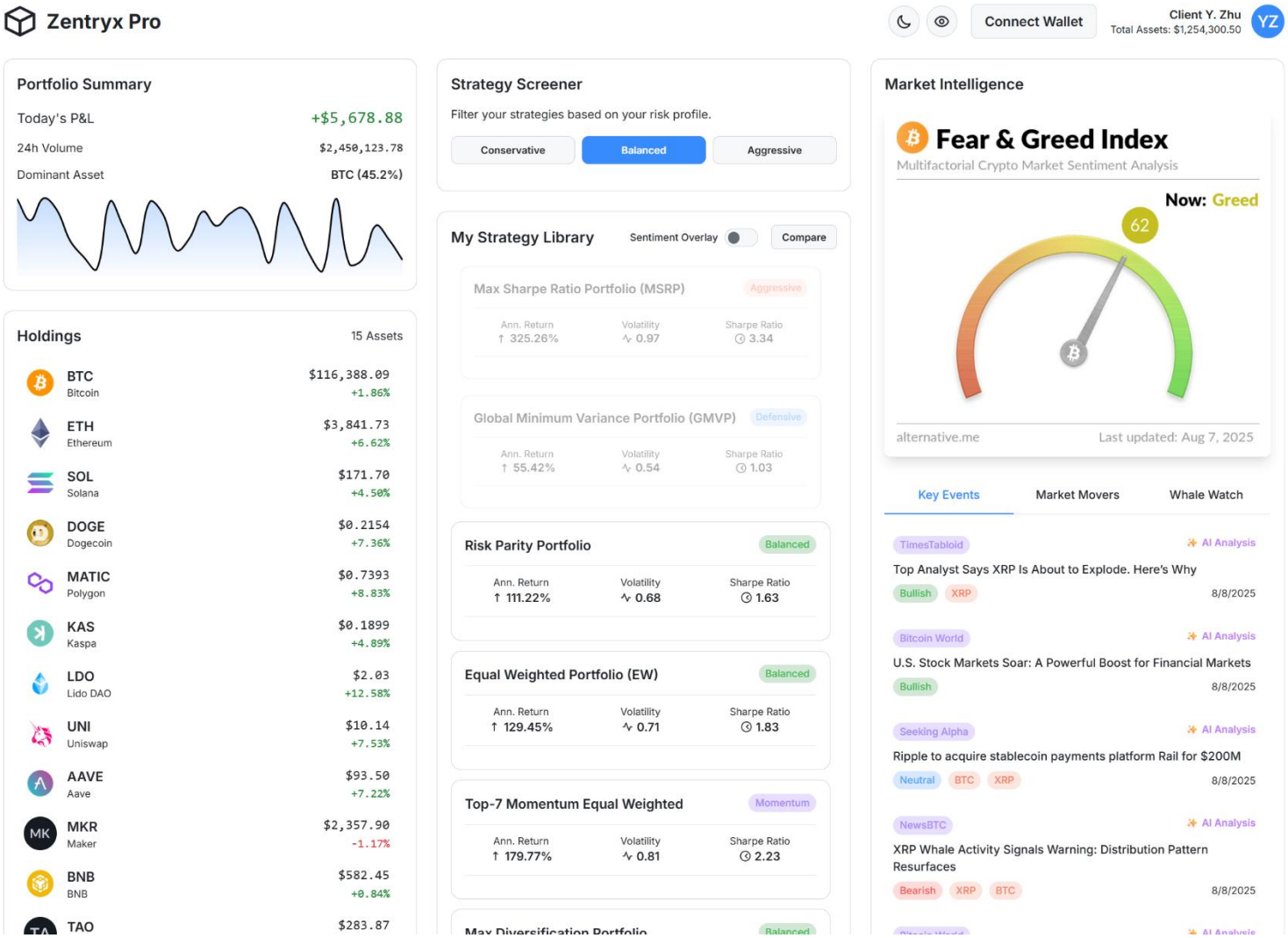


To quantify market sentiment, I implemented a hybrid scoring pipeline which combines VADER and FinBERT using a weighted fusion ($\alpha = 0.5$). VADER handles general sentiment, while FinBERT is used selectively when financial or crypto-specific terms—like “rugpull” or “bullish”, are detected but not scored by VADER. This ensures domain-specific accuracy.

The raw news dataset (from CoinDesk via CryptoCompare) was first cleaned and normalized, then processed into long-format time series with token mentions, sentiment tags, and fused compound scores. These scores were then resampled to weekly and later monthly frequency to match the portfolio rebalancing cycle. Final sentiment indices are stored in a wide-format CSV where each column corresponds to a token's monthly average sentiment.

To validate the signal quality, I performed Granger causality tests between narrative sentiment scores and token returns, identifying significant predictive relationships ($p < 0.05$) for groups such as ETF-sensitive or whale-driven tokens. Additionally, I conducted full backtests with sentiment-tilted strategies. As shown in the SentimentTilted_0.3 dashboard, the strategy achieved a Sharpe Ratio of 2.23 and an annualized return of 191%, outperforming BTC across the entire period. This empirical validation confirms the relevance of the sentiment index in portfolio construction.

4.3 Application development approach incorporating sentiment dashboard

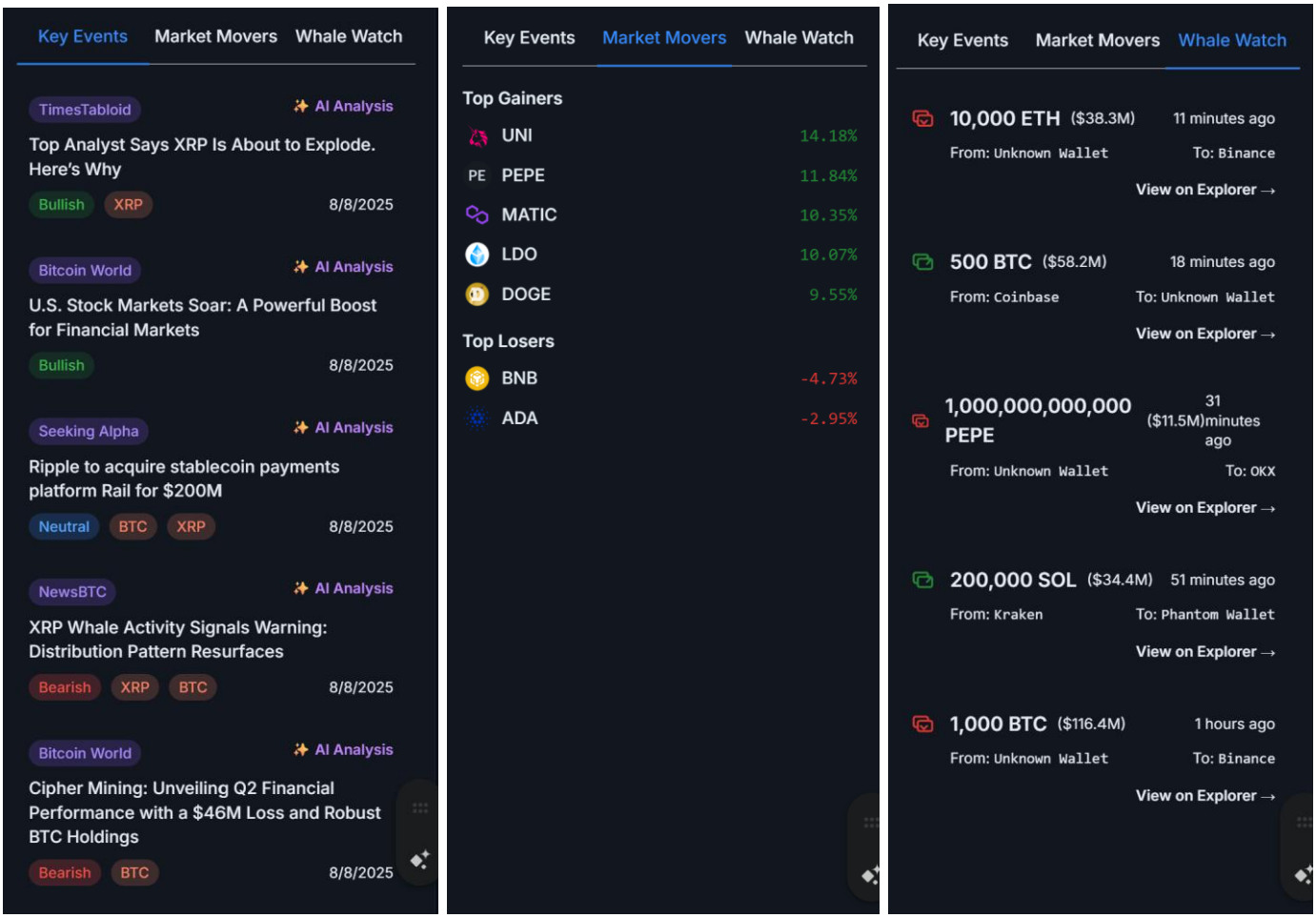


4.4. Google Gemini integration process for enhanced user interaction

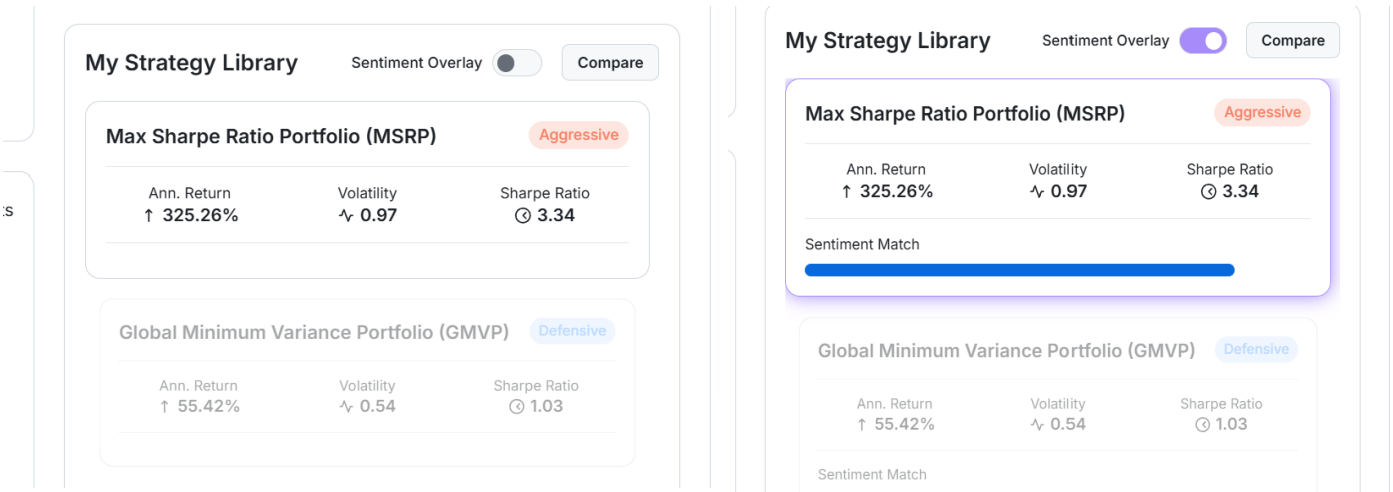
For a better viewing experience, we built an in-house price simulator that automatically refreshes the prices of all assets every few seconds. We run this simulator for extended periods, observing the interface like real users to ensure that price movements and color changes (green for gains, red for losses) flow smoothly, without any lag from frequent updates. At the same time, we monitor the browser console to confirm there are no recurring errors in the loop.

For user interaction, we integrated the live **Fear & Greed Index** from *alternative.me*, using its all-time auto-updating API. The dashboard adapts to any time of day and is accessible for color-impaired users, offering light, dark, and complementary color modes. The layout is designed in three tiers, making it easy to scan: **Key Events**, **Market Moves**, and **Whale Watch**—each with subtitles highlighting details such as the specific coin mentioned in the news, and whether the sentiment is bullish, neutral,

or bearish.



Users can also select their preferred trading style—**conservative, balanced, or aggressive**—and receive a tailored strategy recommendation. For example, in the case shown here, the system recommends MSRP. Additionally, a sentiment filter can be applied to see how market mood impacts the performance of different strategies.



4.5.Testing and debugging procedures for both market and sentiment components

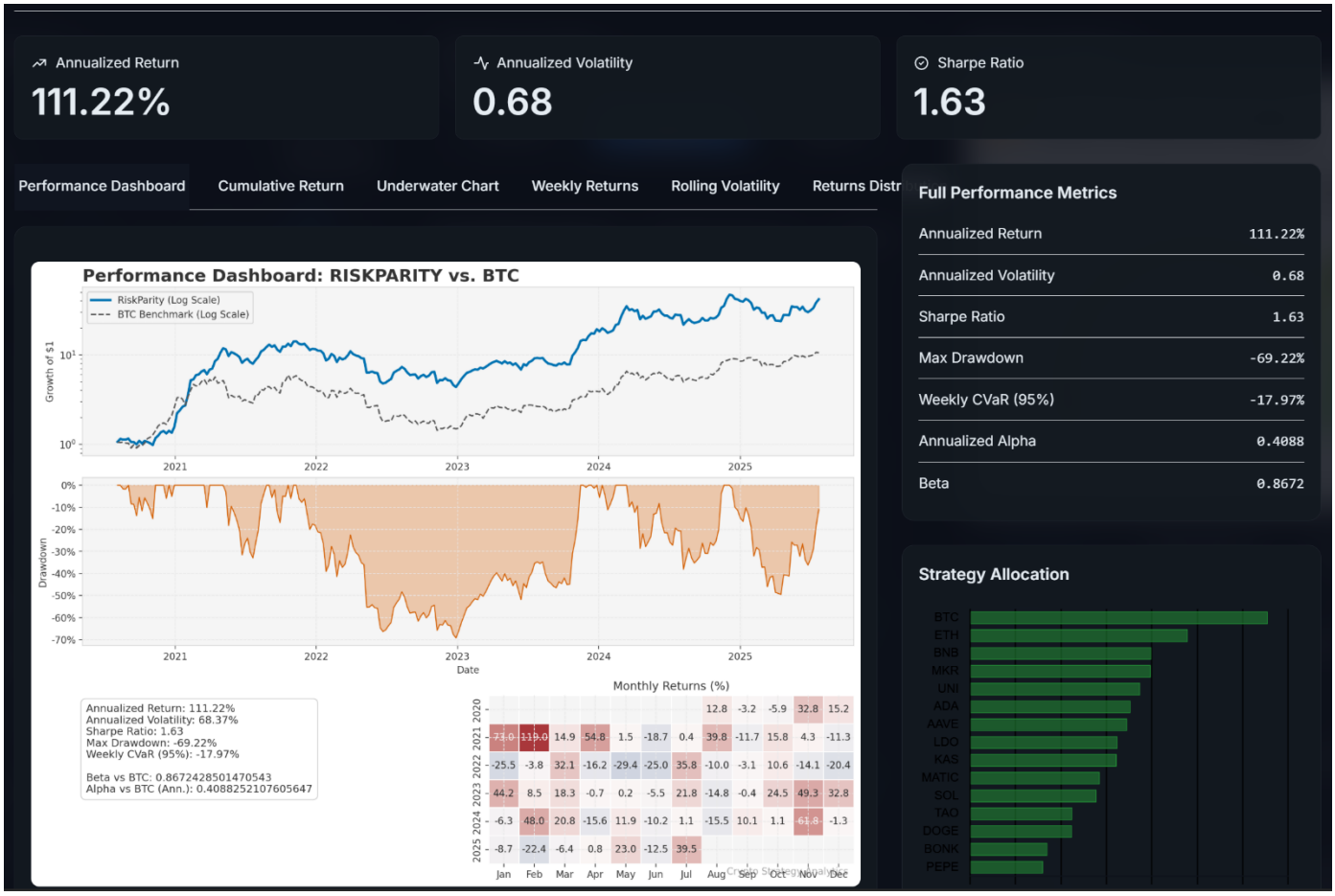
When testing and debugging, I start by checking OHLCV data integrity, filtering out anomalies, and randomly sampling a few days to cross-check against CoinDesk's records. When adding strategies, I also introduce reference variables—such as the alpha parameter in sentiment scoring—and, in cvxpy, enforce diversification in MSRP to avoid over-concentration.

During backtesting, I compare each run's annualized return and Sharpe ratio. If I see an unusually high value, I go back to check the formulas and identify where the calculation went wrong. One early oversight was forgetting that my price data had been weekly resampled, yet I calculated annualized return using a daily frequency. The correct approach was to use a weekly frequency (52 weeks), and this mistake once inflated returns to absurd levels—up to 140,000%.

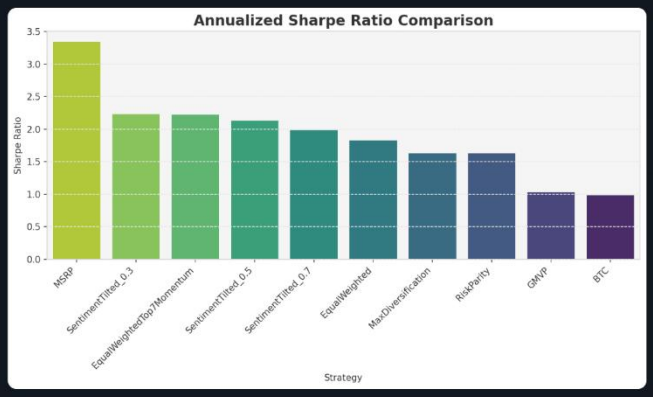
I also experimented with two methods for annualized return: CAGR (Compound Annual Growth Rate) and the Arithmetic Mean. CAGR, because it accounts for compounding, reflects a smoother and more realistic growth rate, as each week's gains become part of the next week's principal. Arithmetic Mean, on the other hand, looked overly optimistic and ignored risk costs, so I ultimately adopted CAGR.

In strategy testing, I initially had a meme constraint rule, capping meme coin exposure at 20% for clients skeptical of them. However, the results were identical to MSRP, so I decided to drop it.

The result showing



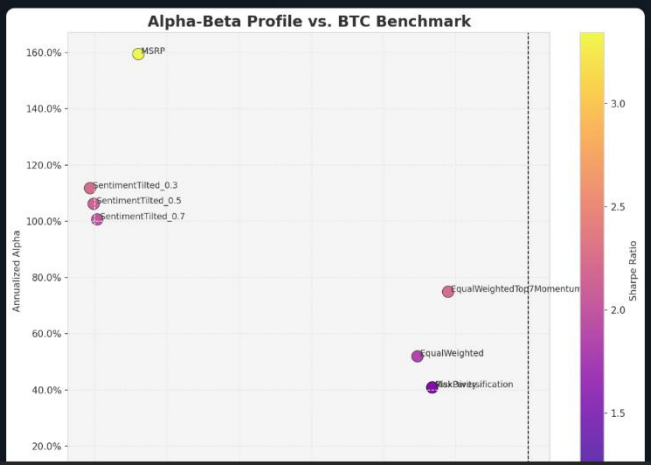
Sharpe Ratio Comparison



Max Drawdown Comparison

Weekly Returns Distribution

Alpha vs BTC Comparison



Reference list

Tiwari, AK, Abakah, EJA, Bonsu, CO, Karikari, NK & Hammoudeh, S 2022, 'The effects of public sentiments and feelings on stock market behavior: Evidence from Australia', Journal of economic behavior & organization, vol. 193, pp. 443–472.

Qu, J & Zhang, L 2023, 'Application of Maximum Sharpe Ratio and Minimum Variance Portfolio Optimization for Industries', Highlights in business, economics and management, vol. 5, pp. 205–213.

Appendix Project Part A

All path are from the source root <5545Crypto_project> + Output of

Station 1:

1. Stage 1 Structured data

Data processing: <stage1_data/crypto_pipeline.py>
Output of <stage1_data/data_stage1_2_result/results/clean_data/stage_1_crypto_data.csv>

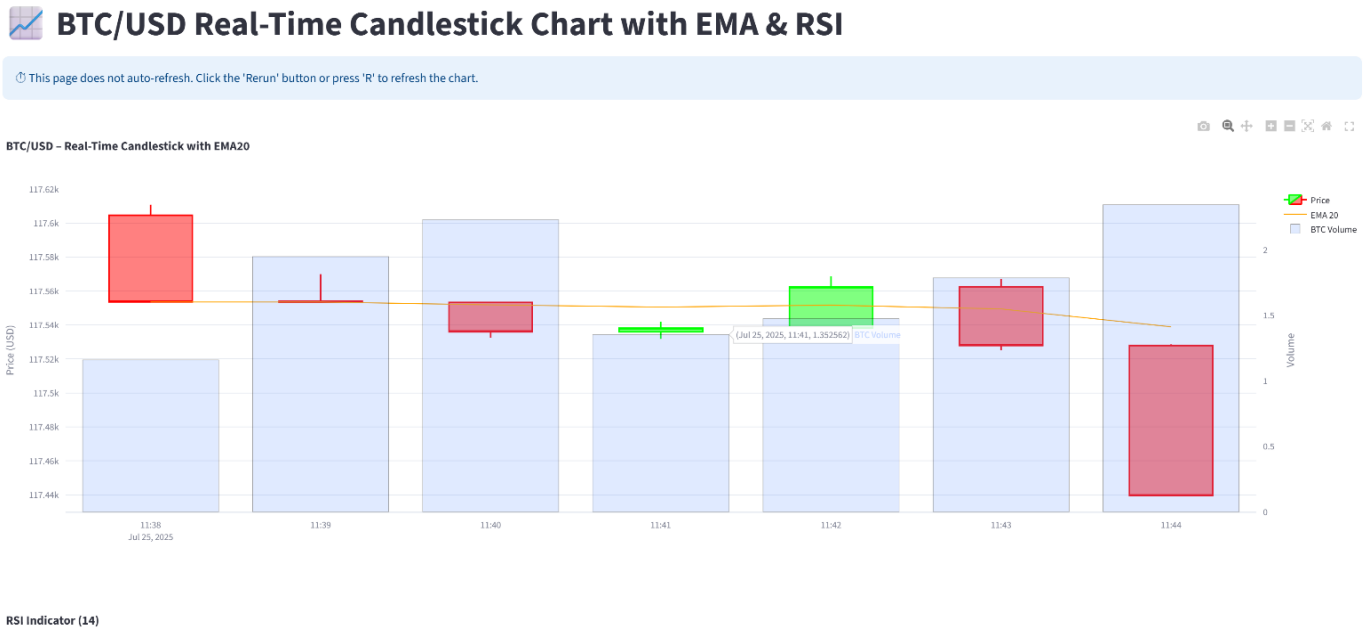
date	id	published_on	title	body	keywords	lang	positive
7/20/2025 14:00	48600435	1753020000	Block: Crypto Mi	Summary Block	XYZ	EN	0
7/20/2025 14:00	48601444	1753020000	Chart of the Wee	Wall Street has	Markets | mar	EN	0
7/20/2025 13:57	48600360	1753019855	Euro debt deals	Emerging-mark	Economy | EU	EN	1
7/20/2025 13:56	48600272	1753019803	XRP price predic	XRP is experien	Cryptocurren	EN	1
7/20/2025 13:54	48600147	1753019650	Bitcoin Price An	Bitcoin (BTC) h	Breaking Nev	EN	0
7/20/2025 13:51	48608392	1753019483	Best Crypto to B	The crypto mar	More News	EN	1
7/20/2025 13:44	48599729	1753019098	Hoskinson prom	Charles Hoskin	Cardano | AD/	EN	0

Real time data catch
Data processing: < stage1_data/realtime_ws_ohlcv.py> and < stage1_data/app_plotly_streamlit.py>
Output of <stage1_data/data/realtime_ohlcv.csv>

Table 1: Sample Real-Time BTC/USD 1-Min OHLCV Data

Symbol	Datetime	Open	High	Low	Close	USD Volume	BTC Volume	USD Vol (mil)
BTC	7/20/2025 8:34	118217.9	118219.9	118217.8	118219.9	2411.01	0.020394	0.002411
BTC	7/20/2025 8:35	118219.9	118221.7	118219.5	118221.7	24921.69	0.210805	0.024922
BTC	7/20/2025 8:36	118221.7	118224.6	118218.0	118218.0	17305.74	0.146388	0.017306
BTC	7/20/2025 8:37	118218.0	118222.7	118215.9	118217.8	5936.0	0.050212	0.005936
BTC	7/20/2025 8:38	118217.8	118253.1	118217.8	118253.1	35535.01	0.300500	0.035535
BTC	7/20/2025 8:39	118253.1	118274.1	118253.1	118274.1	75984.94	0.642448	0.075985

To run the streamlit: cmd C:\Users\Andrea\PycharmProjects\FINS5545\5545Crypto_project\stage1_data>
streamlit run app_plotly_streamlit.py



Unstructured data

Output <stage1_data/data_stage1_2_result/results >

date	id	published_on	title	body	keywords	lang	positive
7/20/2025 14:00	48600435	1753020000	Block: Crypto Mi	Summary Block	XYZ	EN	0
7/20/2025 14:00	48601444	1753020000	Chart of the Wee	Wall Street has	Markets | mar	EN	0
7/20/2025 13:57	48600360	1753019855	Euro debt deals	Emerging-mark	Economy | EU	EN	1
7/20/2025 13:56	48600272	1753019803	XRP price predic	XRP is experien	Cryptocurren	EN	1
7/20/2025 13:54	48600147	1753019650	Bitcoin Price An	Bitcoin (BTC) ha	Breaking New	EN	0
7/20/2025 13:51	48608392	1753019483	Best Crypto to B	The crypto mark	More News	EN	1
7/20/2025 13:44	48599729	1753019098	Hoskinson prom	Charles Hoskin	Cardano | AD/	EN	0

Station 2:

Output of <stage2_features/analyze_volume_jump_impact.py>

<stage2_features/volume_jump_analysis>

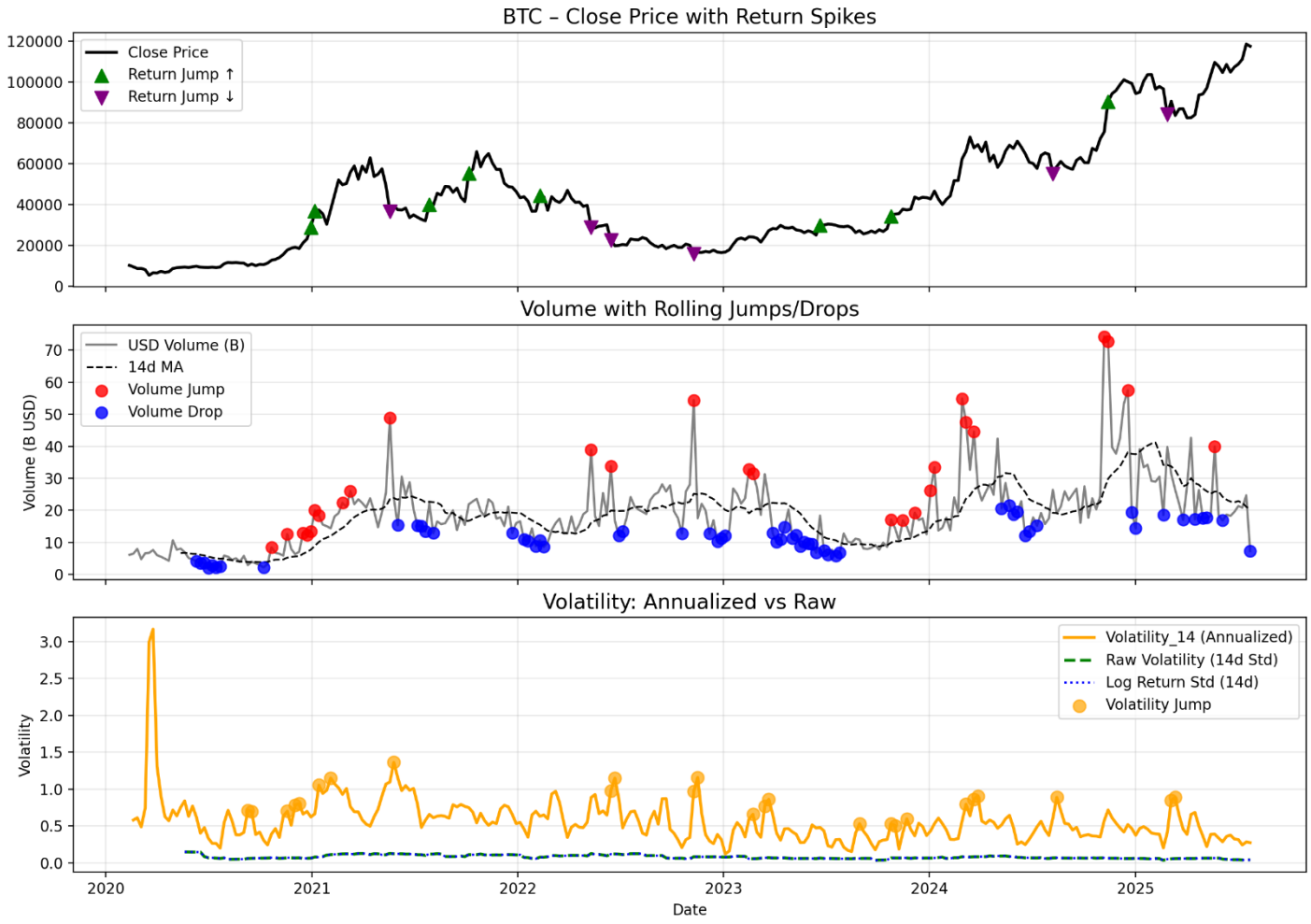
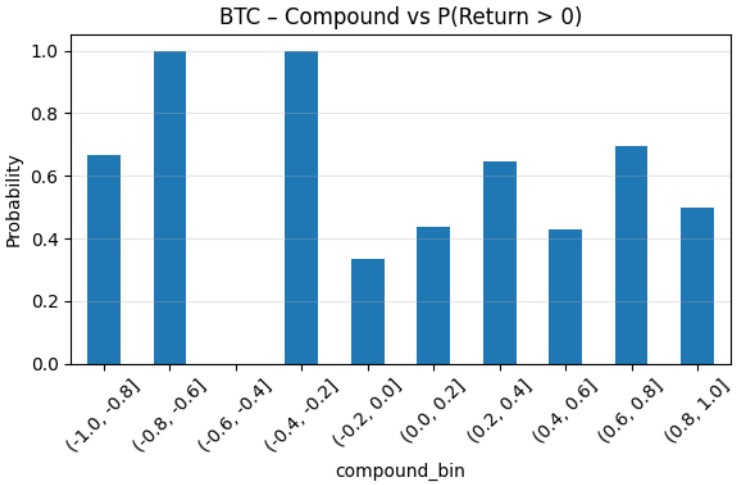


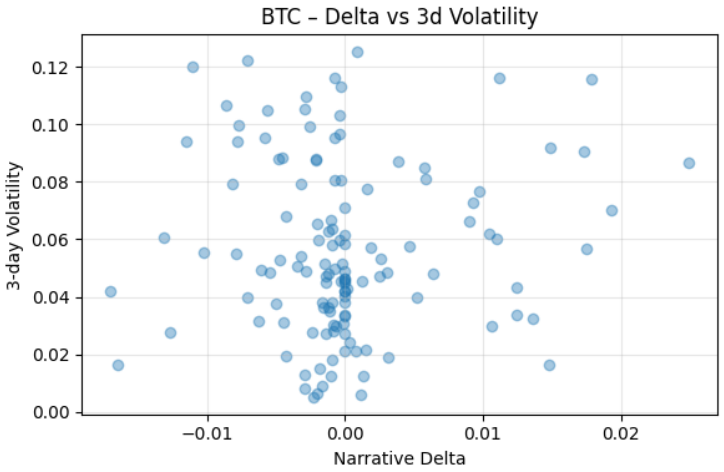
Table 1: BTC Event Impact Summary with Next-Day Returns

Date	Close	Return	Next Rtn	Vol Jump	Vol Drop	Jump ↑	Jump ↓	Volatility Jump
2025-02-26	84197.43	-0.1292	0.0769	False	False	False	True	False
2025-03-05	90671.38	0.0769	-0.0769	False	False	False	False	True
2025-03-12	83701.90	-0.0769	0.0379	False	False	False	False	True
2025-03-26	86939.86	0.0008	-0.0508	False	True	False	False	False
2025-04-16	84057.99	0.0176	0.1150	False	True	False	False	False
2025-04-30	94210.99	0.0052	0.0304	False	True	False	False	False
2025-05-07	97073.38	0.0304	0.0665	False	True	False	False	False
2025-05-21	109686.52	0.0595	-0.0170	True	False	False	False	False
2025-06-04	104729.85	-0.0287	0.0375	False	True	False	False	False
2025-07-23	117574.83	-0.0093	–	False	True	False	False	False

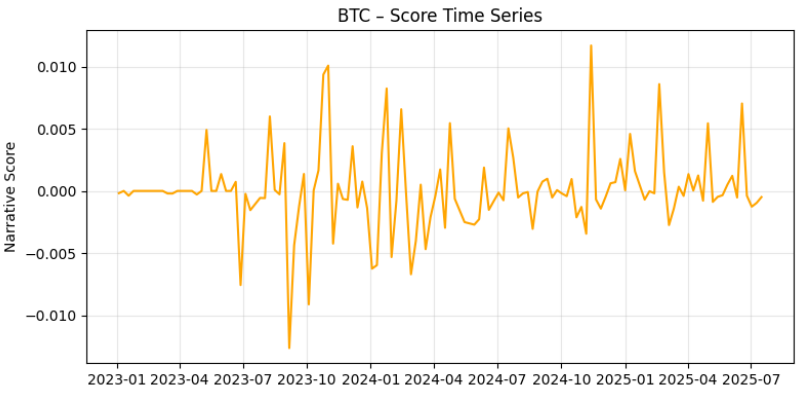
<stage2_features/analyze_volume_jump_impact.py>

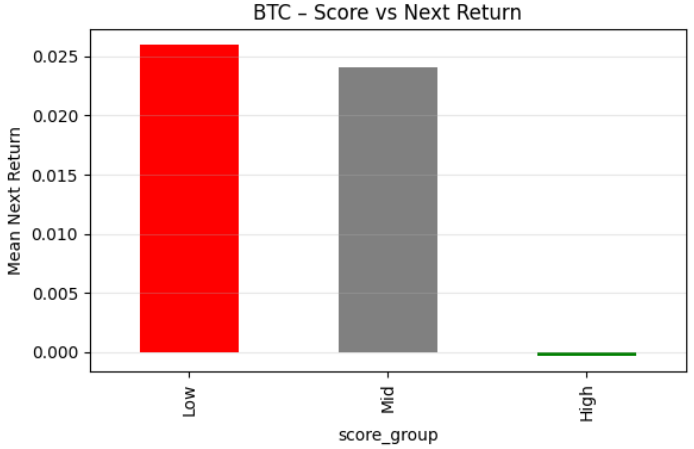
Narrtive analysis output: <stage2_features/narrative_analytics.py>
Example: **BTC & ETF_Event**



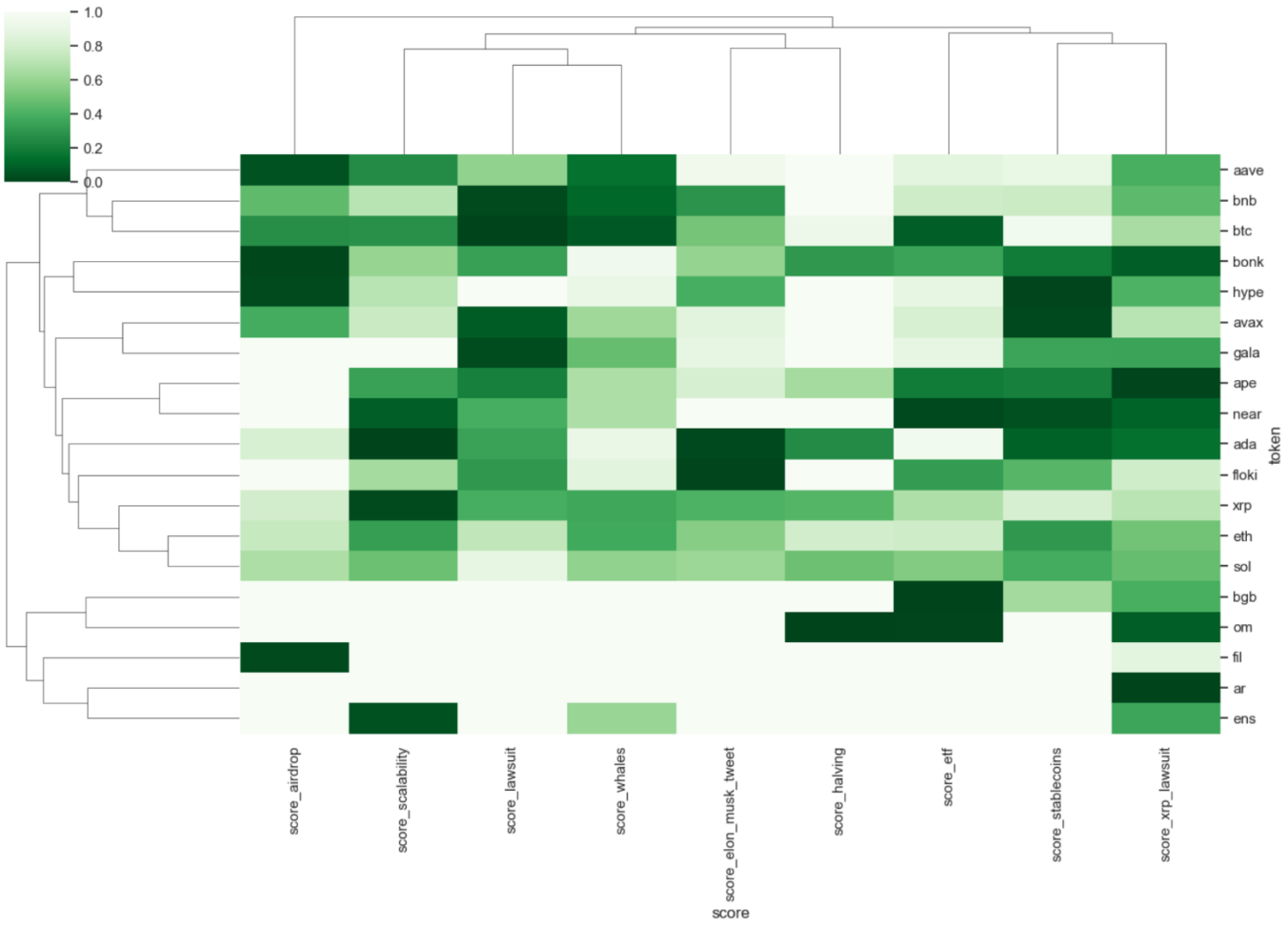


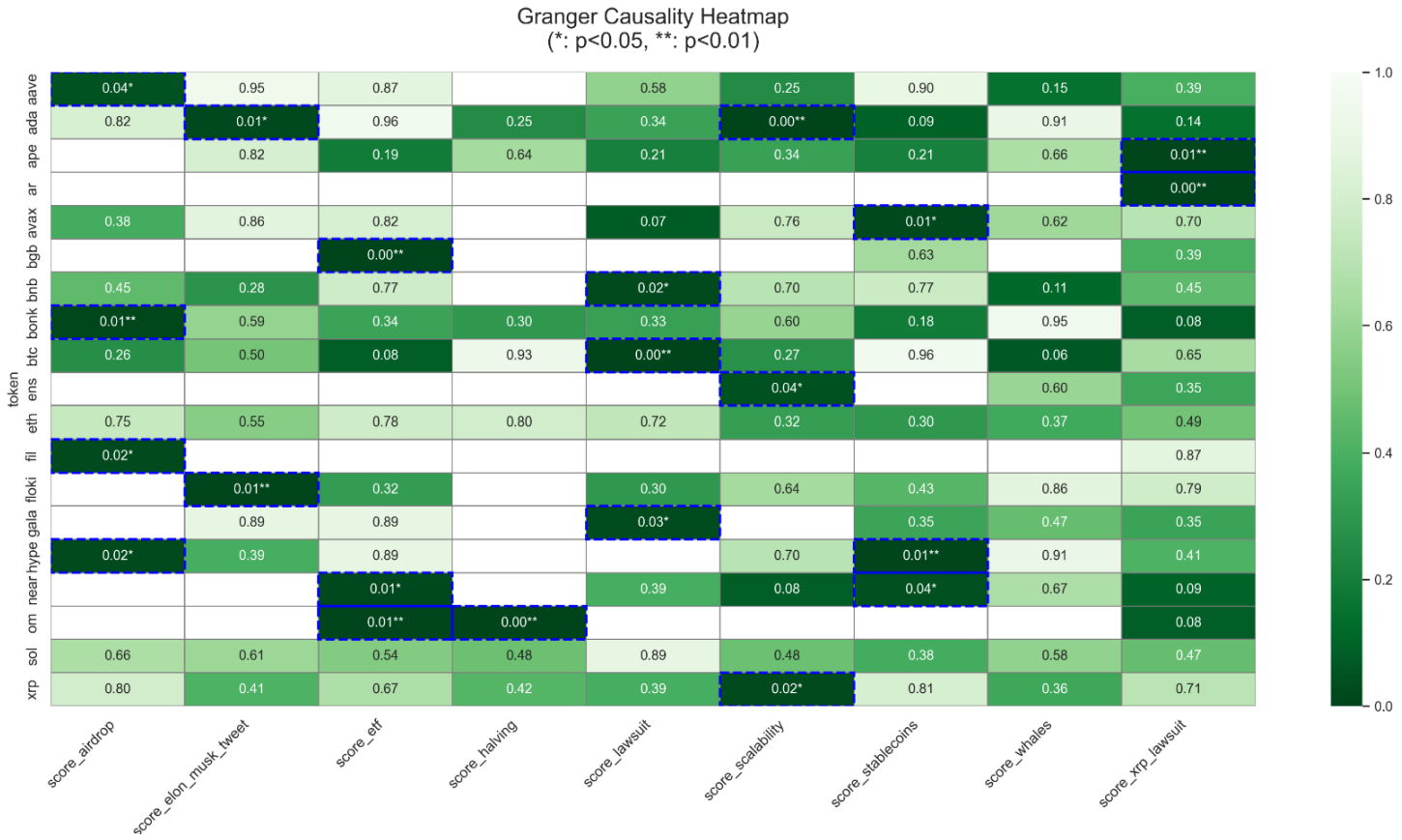
<stage2_features/etf_event/narrative_charts>

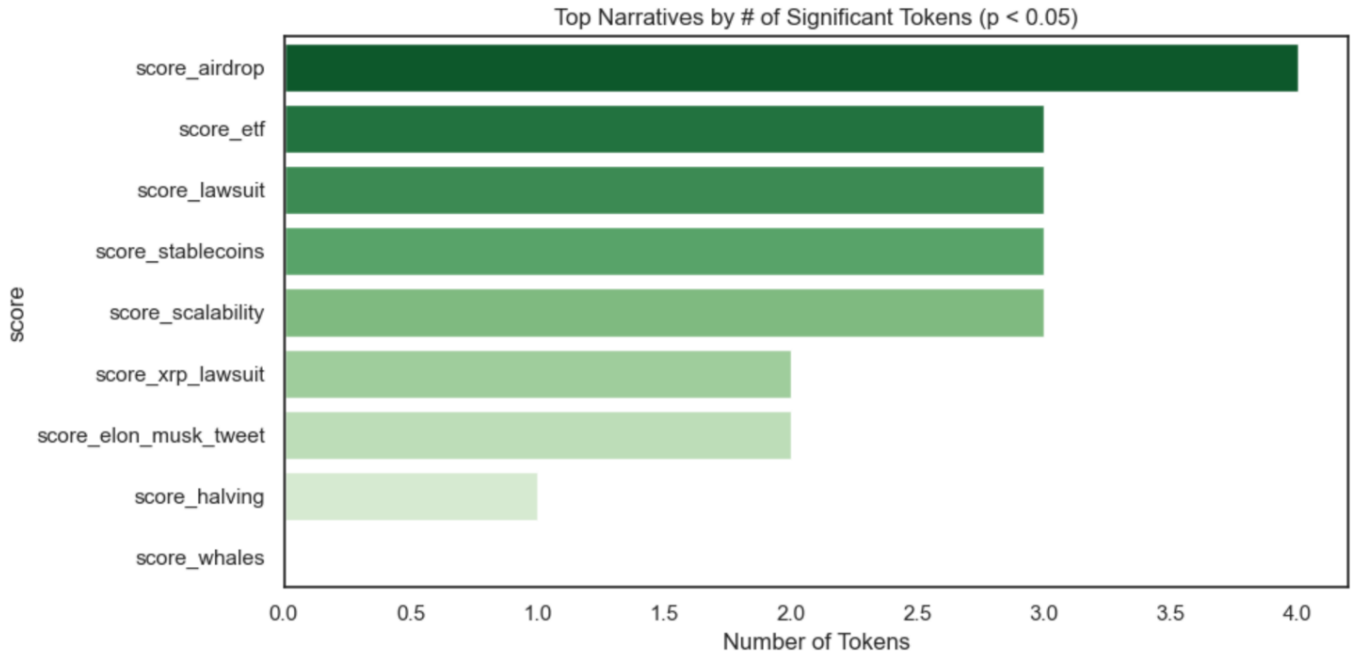




Granger Result <stage2_features/granger_action.py>
Result file: <stage2_features/granger_result>




Granger Causality Heatmap
(*: p<0.05, **: p<0.01)
token
aave 0.04* 0.95 0.87 0.58 0.25 0.90 0.15 0.39
ada 0.82 0.01* 0.96 0.25 0.34 0.00** 0.09 0.91 0.14
ape 0.82 0.19 0.64 0.21 0.34 0.21 0.66 0.01**
ar 0.00**
avax 0.38 0.86 0.82 0.07 0.76 0.01* 0.62 0.70
bgb 0.00** 0.63 0.39
bnb 0.45 0.28 0.77 0.02* 0.70 0.77 0.11 0.45
bonk 0.01** 0.59 0.34 0.30 0.33 0.60 0.18 0.95 0.08
btc 0.26 0.50 0.08 0.93 0.00** 0.27 0.96 0.06 0.65
ens 0.04* 0.60 0.35
eth 0.75 0.55 0.78 0.80 0.72 0.32 0.30 0.37 0.49
fil 0.02* 0.87
floki 0.01** 0.32 0.30 0.64 0.43 0.86 0.79
gala 0.89 0.89 0.03* 0.35 0.47 0.35
hype 0.02* 0.39 0.89 0.70 0.01** 0.91 0.41
near 0.01* 0.39 0.08 0.04* 0.67 0.09
om 0.01** 0.00** 0.08
sol 0.66 0.61 0.54 0.48 0.89 0.48 0.38 0.58 0.47
xrp 0.80 0.41 0.67 0.42 0.39 0.02* 0.81 0.36 0.71
score_airdrop
score_elon_musk_tweet
score_etf
score_halving
score_lawsuit
score_scalability
score_stablecoins
score_whales
score_xrp_lawsuit
1.0
0.8
0.6
0.4
0.2
0.0


Top Narratives by # of Significant Tokens (p < 0.05)
score
score_airdrop
score_etf
score_lawsuit
score_stablecoins
score_scalability
score_xrp_lawsuit
score_elon_musk_tweet
score_halving
score_whales
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Number of Tokens

```
C:\Users\Andrea\PycharmProjects\FINS5545\.venv\Scripts\python.exe C:\Users\Andrea\PycharmProjects\FINS5545\5545Cry
✅ Granger test completed. Results saved to: C:\Users\Andrea\PycharmProjects\FINS5545\5545Crypto_project\stage2_fe
📊 Heatmap saved to: C:\Users\Andrea\PycharmProjects\FINS5545\5545Crypto_project\stage2_features\granger_result\gr
📎 Narrative Cluster Summary:
Theme Cluster 1: includes score_scalability, score_lawsuit, score_whales, score_elon_musk_tweet, score_halving
Theme Cluster 2: includes score_stablecoins, score_xrp_lawsuit
Theme Cluster 3: includes score_etf
Theme Cluster 4: includes score_airdrop

✅ All-token clustering complete. Sample:
   token token_group
0  1inch     Group B
1   aave     Group E
2    ada     Group D
3   aero     Group A
4   aioz     Group B

📊 Token counts per group:
token_group
Group A    77
Group E    26
Group B    16
Group C    10
Group D     8
Group F     1
Name: count, dtype: int64

📊 Token Group Sensitivity Summary (FULL TOKEN SET):
Group A tokens are most sensitive to: score_whales, score_lawsuit, score_xrp_lawsuit
Group B tokens are most sensitive to: score_etf, score_xrp_lawsuit, score_airdrop
Group C tokens are most sensitive to: score_airdrop, score_elon_musk_tweet, score_etf
Group D tokens are most sensitive to: score_elon_musk_tweet, score_scalability, score_stablecoins
Group E tokens are most sensitive to: score_stablecoins, score_whales, score_lawsuit
Group F tokens are most sensitive to: score_halving, score_etf, score_xrp_lawsuit
🔁 Reassigned om from Group F → Group B

Process finished with exit code 0
```

Sentiment: VADER & FinBERT smart score code
Note: the tested VADER & FinBERT result is in week 8 folder

News result: <stage1_data/news_results>
News result after apply VADER & Finbert: <stage1_data/_project_finbert_a05_sentiment_result>

Sentiment pipeline: stage1_data/finbert_a05_sentiment_pipeline.py

Test finbert result is in week 8 folder:
Alpha = 0.3 --→ week8/finbert_a03_week8
Alpha = 0.5 --→ week8/finbert_a05_week8
Alpha = 0.7 --→ week8/finbert_a07_week8
Alpha = 0.1 --→ week8/text_week8
Max weighted method -→ week8/max_weighted_text_week8

```python
# Download VADER lexicon if needed
import nltk
from transformers import pipeline
from nltk.sentiment.vader import SentimentIntensityAnalyzer
import re
import pandas as pd

nltk.download("vader_lexicon")

# Define FinBERT pipeline and cache
_FINBERT = pipeline("sentiment-analysis", model="yiyanghkust/finbert-tone",
tokenizer="yiyanghkust/finbert-tone")
_finbert_cache = {}

# Define finance-related keywords that VADER is insensitive to
FINANCE_KEYWORDS = {
    "bullish", "bearish", "mooning", "rekt", "hodl", "fomo", "fud", "degen",
    "rugpull", "diamondhands", "paperhands", "pump", "dump", "shill",
    "crash", "plummet", "collapse", "drawdown", "correction", "rebound",
    "spike", "whipsaw", "moon", "skyrocket", "surge", "soar",
    "short squeeze", "liquidation", "panic", "fear", "greed",
    "buy the dip", "euphoria", "whale", "weak hands", "strong hands"
}

_VADER = SentimentIntensityAnalyzer()

def _smart_score(txt: str, alpha: float = 0.5) -> pd.Series:
    """
    Hybrid sentiment scoring function combining:
        - VADER (lexicon + rules-based)
        - FinBERT (transformer-based financial context model)

    Adds FinBERT adjustment only if finance-specific slang is detected
    and not covered by VADER lexicon. Uses exponential weighting via alpha.
    Returns: pd.Series with 'neg', 'neu', 'pos', and final 'compound' score.
    """
    vader_scores = _VADER.polarity_scores(txt)
    compound = vader_scores["compound"]

    # Clean text
    clean_text = re.sub(r"[^\w\s]", "", txt.lower())
    tokens = set(clean_text.split())
    present_keywords = tokens & FINANCE_KEYWORDS

    # Check if FinBERT is needed for domain-specific adjustment
    needs_finbert = any(_VADER.lexicon.get(w, 0.0) == 0.0 for w in present_keywords)

    if needs_finbert:
        if txt in _finbert_cache:
            f_score = _finbert_cache[txt]
        else:
            label = _FINBERT(txt[:512])[0]["label"].lower()
            f_score = {"positive": 1.0, "neutral": 0.0, "negative": -1.0}[label]
            _finbert_cache[txt] = f_score

        compound = round(alpha * compound + (1 - alpha) * f_score, 4)

    return pd.Series({
        "neg": vader_scores["neg"],
        "neu": vader_scores["neu"],
        "pos": vader_scores["pos"],
        "compound": compound
    })
eed, [])
}
```

Appendix Project Part B

All the image files please see my imgur gallery

Image files in python:
5545Crypto_project/stage3/crypto_strategy_backtest_full_numpy/backtest_outputs_professional
&
5545Crypto_project/stage3/crypto_strategy_backtest_full_numpy/backtest_outputs_direct_weekly

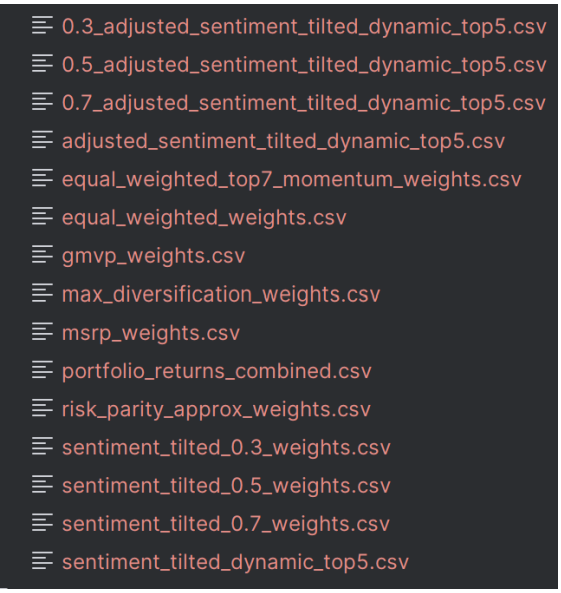
Comparable visualization and dashboard for each strategy: https://imgur.com/gallery/51-MQXKdVY

Each strategy analysis: https://imgur.com/gallery/52-NmJ8trM

Strategy files:

- cvxy_strategy.py (base strategy construction; outputs monthly allocation weights)
- backtest_cvxy_strategy.py (performs backtesting; outputs portfolio_returns_combined.csv)
- portfolio_result.py (visualizes strategy performance; generates strategy_metrics.csv overview for Station 3)
- client_token_result.py (plots individual strategy results across tokens)

Output file:



Allocation csv all in the

5545Crypto_project/stage3/crypto_strategy_backtest_full_numpy

Portfolio returns combined wide format csv is in 5545Crypto_project/stage3/crypto_strategy_backtest_full_numpy/portfolio_returns_combined.csv

	date	MSRP	GMVP	RiskParity	EqualWeighted	Equa
1	2020-08-05	0.134621	0.028200	0.065388	0.053943	
2	2020-08-12	0.201634	0.033116	0.077074	0.091546	
3	2020-08-19	0.001906	-0.003399	-0.016491	-0.020905	
4	2020-08-26	0.093584	-0.003795	-0.000639	0.005950	
5	2020-09-02	0.101864	-0.012650	0.023856	0.017475	
6	2020-09-09	0.023032	-0.077766	-0.084726	-0.073893	
7	2020-09-16	0.014862	0.052769	-0.001480	-0.010836	
8	2020-09-23	-0.021827	-0.044920	-0.056039	-0.045774	
9	2020-09-30	0.061477	0.056542	0.095925	0.082443	

Strategy metric output: stage3/crypto_strategy_backtest_full_numpy/backtest_outputs_professional/strategy_metrics_summary.csv

	<anonymous>	Annualized Return	Annualized Volatility	Sharpe Ratio	Max Drawdown	Weekly CVaR (95%)	A
1	MSRP	3.2526	0.9724	3.3448	-0.7067	-0.1942	
2	GMVP	0.5542	0.5360	1.0338	-0.7657	-0.1572	
3	RiskParity	1.1122	0.6837	1.6268	-0.6922	-0.1797	
4	EqualWeighted	1.2945	0.7076	1.8294	-0.6589	-0.1752	
5	EqualWeightedTop7Momentum	1.7977	0.8077	2.2258	-0.6192	-0.1698	
6	MaxDiversification	1.1122	0.6837	1.6268	-0.6922	-0.1797	
7	SentimentTilted_0.3	1.9107	0.8561	2.2319	-0.5954	-0.1438	
8	SentimentTilted_0.5	1.7833	0.8367	2.1314	-0.5997	-0.1537	
9	SentimentTilted_0.7	1.6437	0.8270	1.9876	-0.6046	-0.1650	
10	BTC	0.6028	0.6094	0.9892	-0.7590	-0.1707	

Monthly sentiment score for the selected coins: stage3/monthly_sentiment.csv

	date	BTC	ETH	SOL	DOGE	MATIC	KAS	LDO	UNI	AAVE	MKR
1	2023-01-04	0.280913	0.320136	0.488814	0.601100	0.242258	-0.144400	0.764200	0.502297	<null>	-
2	2023-02-01	0.464910	0.558169	0.548620	0.611749	0.532909	0.999500	0.385458	0.537381	0.187240	
3	2023-03-01	0.249998	0.428130	0.190983	0.484683	0.556787	0.896433	0.141783	0.479568	-0.243025	
4	2023-04-05	0.400471	0.351849	0.389530	0.356158	0.435520	-0.487100	0.735100	0.514283	0.631000	
5	2023-05-03	0.380383	0.488528	0.354642	0.366655	0.596349	0.989800	0.997750	0.549478	0.632300	
6	2023-06-07	0.443229	0.510218	0.400443	-0.142958	0.697531	<null>	0.298742	0.549090	0.760500	
7	2023-07-05	0.509997	0.585179	0.581900	0.536746	0.813366	0.996500	0.582783	0.543506	0.401900	
8	2023-08-02	0.334115	0.504057	0.512446	0.605244	0.517980	<null>	0.727100	0.564193	-0.496550	
9	2023-09-06	0.294123	0.431545	0.581736	0.193046	0.214007	<null>	0.683400	0.457105	<null>	