

Crypto-Media Integration: A NoSQL Data Pipeline for Market Sentiment Analysis

Data Management Project Report

Tommaso Arone ID: 896282

Lorenzo Zanotti ID: 902652

Lorenzo Triolo ID: 895541

January 11, 2026

Abstract

This project investigates whether mainstream media coverage can be linked to cryptocurrency market behavior. A complete data-management pipeline is implemented, combining (i) daily market data for major assets and (ii) news metadata extracted via the New York Times Article Search API. Data are acquired from heterogeneous sources, transported through a streaming layer, stored in a document-oriented database, integrated on a daily time axis, and assessed with explicit data-quality metrics.

Project snapshot

- **Goal.** Measure the relationship between media attention and crypto market dynamics.
- **Core idea.** Aggregate news by day (news volume) and align it with daily market indicators.
- **Stack.** Python (requests, pandas), Kafka, MongoDB, Jupyter.

1 Introduction and Objectives

Cryptocurrency markets are known for their high reactivity to external information, such as regulatory announcements and institutional adoption. However, information flow is difficult to measure directly; this project utilizes mainstream media coverage as a quantifiable proxy, specifically analyzing the impact of The New York Times articles on Bitcoin (BTC) and Ethereum (ETH) market behaviors.

In strict compliance with the project requirements, the work is structured around three core objectives:

- **Data Acquisition and NoSQL Storage:** The project implements a scalable pipeline that harvests data from two distinct sources using APIs (*New York Times Article Search API* and *Yahoo Finance*). To handle the heterogeneity of the data—unstructured text from articles and structured time-series from financial markets—the dataset is integrated and persisted in a MongoDB (NoSQL) database using a de-normalized document structure designed for analytical queries.
- **Data Quality Assessment and Improvement:** Before analysis, the raw data and the integrated dataset undergo a rigorous quality assessment focusing on three key dimensions: *Uniqueness* (resolving duplication artifacts resulting from multi-query API ingestions), *Completeness* (imputing missing textual fields and quantifying temporal "blind spots" where market data

exists without corresponding news), and *Semantic Consistency* (measuring information redundancy via cosine similarity to detect repetitive narratives). Specific cleaning procedures and metrics are applied to ensure the reliability of the dataset for downstream tasks.

- **Exploratory Data Analysis (EDA):** Finally, the project performs an integrated exploratory analysis on the curated dataset. The goal is to evaluate correlations between media attention (news volume) and market indicators (volatility and price returns), demonstrating the pipeline's ability to support insight generation from diverse data sources.

2 Research Questions

1. **RQ1.** Is there an association between NYT news volume about cryptocurrencies and Bitcoin/Ethereum price/volume?
2. **RQ2.** Does media attention behave as a *leading* indicator for market movements (e.g., next-day or next-week changes)?

3 Datasets and Data Acquisition

The data acquisition layer is designed to harvest heterogeneous information from two distinct sources, combining structured financial time-series with unstructured textual metadata.

3.1 Financial Market Data (Structured)

The backbone of the quantitative analysis is provided by historical daily market data retrieved via the *yfinance* Python library, which interfaces with Yahoo Finance. The dataset focuses on two major cryptocurrencies and a traditional market index for potential correlation benchmarking:

- **Bitcoin (BTC-USD)**
- **Ethereum (ETH-USD)**

For each asset, data is ingested at a daily granularity. The schema includes standard OHLCV metrics: *Open*,

High, *Low*, *Close* prices, and *Volume*. This structured dataset provides the temporal consistency required for the subsequent integration phase, serving as the primary index for the daily buckets in the NoSQL database.

3.2 News Media Data (Unstructured)

To quantify media attention, the project utilizes the **New York Times Article Search API**. This source was selected due to the publication's high reputation and its influence on mainstream investor sentiment. The acquisition pipeline iterates through three specific query terms to capture a broad spectrum of the crypto-ecosystem coverage:

- "bitcoin"
- "ethereum"
- "crypto"

Data Schema For each retrieved article, the system filters and stores a compact JSON document containing essential metadata for sentiment analysis and identification:

- *Publication Date* (normalized to UTC)
- *Headline* and *Abstract* (for textual analysis)
- *Web URL* (for lineage and verification)
- *Word Count* (to filter out brief snippets)
- *Source Query* (to track which keyword triggered the result)

Ingestion Logic and Rate Limiting The NYT API imposes strict rate limits (typically limited to a specific number of requests per minute). To ensure robust data ingestion without data loss, the Python script implements a defensive pagination strategy. The extraction is segmented month-by-month and iterates through result pages (0 to 100). The pipeline includes exception handling for HTTP 429 (Too Many Requests) errors, implementing a “sleep-and-retry” mechanism to respect the API’s constraints automatically.

3.3 News Data Acquisition Implementation

The acquisition of unstructured news data is implemented through a decoupled **Producer-Consumer architecture** utilizing **Apache Kafka** as the intermediate message broker. This design ensures modularity and prevents data loss in case of database downtime. The workflow is divided into three distinct stages:

3.3.1 The Producer: Ingestion and Rate Limiting

The ingestion script (Kafka Producer) is responsible for querying the *NYT Article Search API*. The extraction covers the timeframe from **January 1, 2018** to **December 12, 2025**. To manage the API’s constraints and the pagination logic effectively, the script operates via a nested iteration strategy:

1. **Temporal Segmentation:** The time window is iterated month-by-month to respect the API’s pagination

limits (maximum 100 pages per query/filter).

2. **Topic Iteration:** For each month, the script cycles through the three target keywords: "Bitcoin", "Crypto", and "Ethereum".
3. **Defensive Pagination:** The script iterates through result pages (0–100). To avoid hitting the API’s “Too Many Requests” error, a conservative delay of **12 seconds** is enforced between standard calls.

Fault Tolerance and Limits The implementation includes robust error handling mechanisms:

- **HTTP 429 Handling:** In the event of an API rate-limit violation (HTTP 429), the script automatically pauses execution for 70 seconds before retrying the failed page.
- **Daily Safety Cap:** A strict local limit of **DAILY_LIMIT = 480** calls is implemented to prevent exceeding the daily quota (500) provided by the API key, reserving a buffer for debugging operations.

Before transmission, raw JSON responses are filtered to extract only relevant fields (e.g., *headline*, *abstract*, *pub_date*, *web_url*), creating a **clean_doc** object. This lightweight object is then serialized and sent to a specific Kafka topic corresponding to the query keyword.

3.3.2 The Consumer: Persistence to NoSQL

The downstream component is a set of **Kafka Consumers**, one for each topic (**Bitcoin**, **Crypto**, **Ethereum**). These consumers subscribe to the message broker using the **earliest** offset to ensure all historical data is captured. Upon receiving a message:

1. The JSON payload is deserialized.
2. The document is enriched with a *query* tag to preserve the originating search term context.
3. The final document is inserted into the **MongoDB** collection **DM_DB.Article**.

This architecture ensures that the ingestion process (which is network-bound and slow due to rate limits) is decoupled from the storage process, allowing for asynchronous data processing.

3.4 Financial Data Processing and Persistence

Unlike the asynchronous pipeline used for news data, the financial market data is processed via a direct ETL (Extract-Transform-Load) script. The raw data retrieved from **yfinance** is initially structured in a multi-level column format ("Wide" format). To make this data suitable for document-based storage and querying, the pipeline performs the following transformation and cleaning steps:

1. **Reshaping (Wide-to-Long):** The script utilizes the Pandas **stack** method to pivot the dataset. This transforms the structure from having tickers as columns to a "Tidy" format where each row represents a single observation (Asset, Date, Price). The index is then reset to treat **Date** and **Ticker** as standard fields.
2. **Schema Normalization:** Column names are standardized to follow a consistent naming convention

- (snake_case, lowercase) to ensure uniformity across the database (e.g., converting "Adj Close" to `adj_close`).
3. **Type Enforcement:** The date field is explicitly cast to a Datetime object to enable time-based operations within MongoDB.
 4. **NoSQL Compatibility (Null Handling):** A crucial step involves handling missing values. Since Pandas represents missing numeric data as `NaN` (Not a Number), which is not natively supported by standard JSON/BSON serialization, the script explicitly maps all `NaN` values to Python `None`. This ensures they are correctly stored as BSON `null` types in the database.

Persistence Once transformed, the DataFrame is converted into a list of dictionary records. These records are persisted into the **MongoDB** collection `DM_DB.Crypto` using a bulk write operation (`insert_many`). This approach minimizes network overhead compared to inserting documents one by one.

4 Storage Design and Data Integration Strategy

To support the research objective, analyzing the correlation between media attention and cryptocurrency market volatility, the storage architecture was implemented using **MongoDB**. This choice was driven by specific requirements regarding data heterogeneity and query patterns.

4.1 Rationale for MongoDB Selection

The decision to adopt a NoSQL document-oriented database over a traditional Relational Database Management System (RDBMS) is justified by three architectural advantages relevant to this case study:

- **Handling Semi-Structured Data:** News metadata retrieved from the NYT API are inherently polymorphic (e.g., varying abstract lengths, missing fields, diverse keyword tags). MongoDB's flexible schema allows for the ingestion of these JSON documents without enforcing rigid tables or complex migrations when source formats change.
- **Data Locality via Embedding:** The analysis requires accessing all relevant information for a specific date (prices and news) simultaneously. By using the *Embedded Data Model*, we store the list of daily articles directly inside the daily market document. This eliminates the need for expensive JOIN operations between an "Articles" table and a "Prices" table, significantly reducing read latency during the exploratory analysis phase.
- **Time-Based Bucketing:** The data model is designed around the concept of a "Daily Bucket". Each document represents a single day, aggregating diverse data sources into a unified view. This aligns perfectly with the research question, which investigates daily and weekly correlations.

4.2 Integration Logic: The Daily Market Summary

The integration pipeline (implemented in Python) transforms the raw staging collections (`Article` and `Crypto`) into a final, analytics-ready collection named `daily_market_summary`. The integration process follows a "Materialized View" strategy, structured in four phases:

1. **Temporal Normalization:** To ensure alignment between the New York Times publication times and financial market closing times, all timestamps were normalized to UTC midnight. This resolved potential inconsistencies due to timezone differences (Jet Lag issue).
2. **Data Aggregation (Group-By):** Articles were grouped by their normalized publication date. A Python dictionary structure was used to map each date to its corresponding list of cleaned article objects (containing only relevant fields like `headline`, `abstract`, and `source_query`).
3. **Contextual Enrichment (The Look-Ahead Mechanism):** To directly address the research question regarding predictive power, the integration script does not merely copy price data; it performs a *calculation of trends*. For every specific date t_0 , the system looks up prices for:

- t_{-1} (Previous day) to calculate context.
- t_{+1} and t_{+7} (Next day and Next week) to calculate **future trends**.

This effectively "materializes" the future deltas into the current document. For example, a document for Jan 24th contains the field `future_trend.next_week_diff`, which is the price difference between Jan 31st and Jan 24th.

4. **Persistence:** The integrated documents are stored in the `DM_DB_Integrated` database. This separation between *Source DB* (Raw) and *Destination DB* (Integrated) ensures the pipeline is idempotent and non-destructive.

4.3 Final Data Model

The resulting document structure in the `daily_market_summary` collection is denormalized to optimize analytical queries. A typical document follows this schema:

```
{
  "_id": {
    "$oid": "694588d1d65f26c5a42ef2f8"
  },
  "date": {
    "$date": "2018-01-24T00:00:00.000Z"
  },
  "date_str": "2018-01-24",
  "articles": [
    {
      "remote_id": "...",
      "title": "...",
      "url": ...
    }
  ]
}
```

```

    "source_query": "Bitcoin",
    "abstract": "..."
},
{
  "remote_id": "...",
  "title": "...",
  "url": "...",
  "source_query": "Bitcoin",
  "abstract": "..."
}
],
"article_count": 2,
"prices": {
  "BTC-USD": {
    "open": 10903.400390625,
    "close": 11359.400390625,
    "intraday_change": 456,
    "context": {
      "close_prev_day": 10868.400390625,
      "close_next_day": 11259.400390625,
      "close_prev_week": 11188.599609375,
      "close_next_week": 10221.099609375
    },
    "future_trend": {
      "next_day_diff": -100,
      "next_week_diff": -1138.30078125
    }
  },
  "ETH-USD": {
    "open": 987.4769897460938,
    "close": 1058.780029296875,
    "intraday_change": 71.30303955078125,
    "context": {
      "close_prev_day": 986.22900390625,
      "close_next_day": 1056.030029296875,
      "close_prev_week": 1014.25,
      "close_next_week": 1118.31005859375
    },
    "future_trend": {
      "next_day_diff": -2.75,
      "next_week_diff": 59.530029296875
    }
  }
}
}

```

This structure allows analysts to retrieve the complete market context and media sentiment for any given timeframe with a single MongoDB query, satisfying the project's performance and usability requirements.

5 Query Example:

The following queries serve as a benchmark for evaluating the database's performance in high-frequency data management tasks. They demonstrate the schema's ability to seamlessly correlate time-series financial data with unstructured news metadata.

5.1 Query 1: High-Volume News Correlation

Query 1 (Listing 1) identifies days with the highest density of news coverage. By sorting the dataset by `article_count` in descending order, we can observe whether peak media attention coincides with significant price shifts, such as during the FTX collapse. The corresponding results are summarized in Table 1.

5.2 Query 2: Crash and Recovery Analysis

Query 2 (Listing 2) utilizes conditional `$switch` logic to categorize market resilience. It filters for days where Bitcoin experienced significant intraday drops and evaluates the recovery type: "V-Shape" for rapid 24-hour rebounds or "U-Shape" for slower weekly recoveries. The performance analysis is shown in Table 2.

5.3 Query 3: The "News-First" Collapse

Query 3 (Listing 3) targets the "Next Day" effect. It identifies instances where news articles preceded major 24-hour price drops. This allows for an investigation into whether specific headlines acted as immediate market catalysts, as detailed in Table 3.

5.4 Query 4: Macro Volatility Baseline

To assess the overall impact of information flow, Query 4 (Listing 4) groups the entire dataset into two categories. By establishing a statistical baseline (Table 4), we confirm that market volatility for both BTC and ETH is significantly higher on days accompanied by media coverage.

5.5 Query 5: The "Blind Spot" Analysis

The most sophisticated pipeline, Query 5 (Listing 5 and 6), targets "Information Vacuums"—days with 0 articles but extreme volatility. Using a double `$lookup`, it scans for articles in the 3-day windows surrounding the move to determine if the market was reacting to delayed or hidden news. The findings are presented in Table 5.

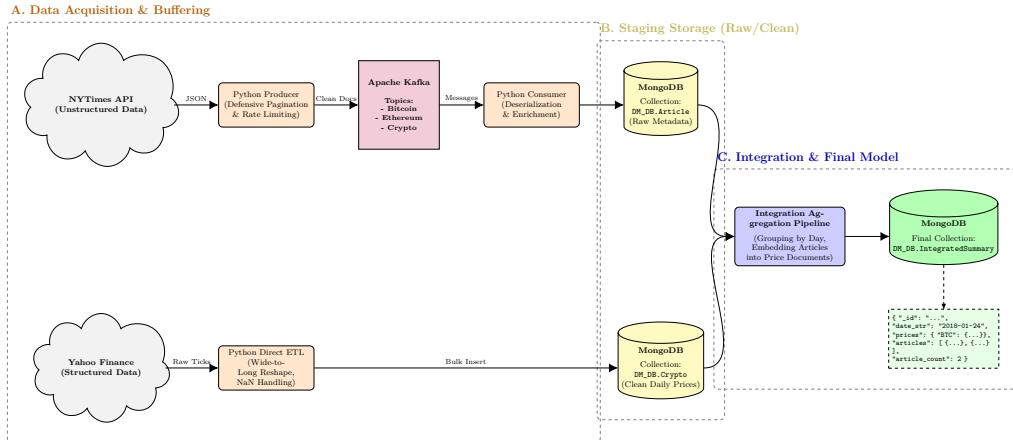


Figure 1: Schema Architecture

Table 1: Results for Query 1: Days with Highest Article Count

Date	#Arts	Top News Title	BTC Close	BTC Vol.	ETH Close	ETH Vol.
2022-12-13	15	In FTX Collapse...	17,781.32	+574.88	1,320.55	+45.89
2022-11-30	13	Before FTX fell...	17,168.57	+723.09	1,295.69	+78.76
2018-06-28	13	A Field Guide...	5,903.44	-249.72	422.36	-19.93
2021-05-13	12	Elon Musk Makes...	49,716.19	-19.24	3,715.15	-113.77
2021-04-14	12	Coinbase Listing...	63,109.70	-414.06	2,435.10	+135.76

Table 2: Query 2: Market Crash and Recovery Classification

Date	Arts	BTC Crash	BTC 24h	BTC 7d	BTC Rec.	ETH Rec.
2025-11-17	11	-13,902.72	+855.00	-3,823.31	V-Shape	V-Shape
2021-05-18	4	-13,795.17	-5,906.96	-4,507.18	No Rec.	No Rec.
2025-03-09	1	-13,647.31	-2,069.04	+1,978.65	U-Shape	No Rec.
2024-08-05	2	-12,828.46	+2,042.86	+5,363.06	V-Shape	V-Shape
2021-05-21	3	-12,575.84	+231.94	-1,607.09	V-Shape	No Rec.

Table 3: Query 3: Top Price Collapses After News

Date	News	Primary Article Title	BTC Close	BTC Next Day	BTC Wkly
2025-10-09	1	Crypto Investor Known as...	121,705.59	-8,491.22	-13,519.55
2025-03-02	1	Is 'Wicked' Really a...	94,248.35	-8,182.68	-13,647.31
2021-05-11	1	L Brands Plans to Spin Off...	56,704.57	-7,554.04	-13,795.17
2024-12-17	3	Late Night Delights in...	106,140.60	-6,099.06	-7,464.51
2021-05-18	4	Elon Musk Impostors...	42,909.40	-5,906.96	-4,507.18

Table 4: Query 4: Avg Volatility (News vs. No News)

Category	Days	BTC Avg Abs Vol	ETH Avg Abs Vol
Days WITH News	1632	982.91	65.52
Days WITHOUT News	1270	474.03	28.65

Table 5: Query 5: Volatility in Blind Spots

Date	Pre-3d	Post-3d	BTC Daily	BTC Abs. Vol	ETH Daily
2021-11-26	5	3	-5,390.52	5,390.52	-491.30
2025-04-06	2	8	-5,290.02	5,290.02	-229.23
2025-11-04	7	8	-4,950.90	4,950.90	-309.47
2024-05-15	11	2	+4,713.50	4,713.50	+155.83
2025-06-23	1	5	+4,590.30	4,590.30	+193.34

6 Data Quality Assessment

The project computes:

- **Articles.** Missing abstract rate and duplicate rate.
- **Advanced textual checks.** A redundancy rate based on string similarity (cosine similarity), aimed at detecting redundancy in days with more than one article.
- **Aggregated dataset.** Blind spots, i.e., days with market data but zero news items.

6.1 Duplication Rate

The NYT Article Search API ingestion initially produced 7257 documents. A duplication analysis based on repeated URLs revealed a duplication rate of 27.697%, indicating that the same content was retrieved multiple times. After removing 3592 duplicated records, the final staging collection contains 3665 unique articles.

This phenomenon is consistent with the multi-query acquisition design: the keywords used in the NYT queries (e.g., “Bitcoin”, “Crypto”, “Ethereum”) may match the same article across different requests, generating duplicated URLs and repeated identifiers.

6.2 Completeness

Articles. We assessed completeness on core textual fields used for downstream content analysis. Although titles are consistently available, 4.18% of articles exhibited a missing (null or empty) abstract. Since the abstract is required for topic and sentiment extraction, we applied a conservative imputation policy:

```
abstract ← title if abstract is missing.
```

After this correction, the dataset presents 0% missing abstracts, ensuring full textual availability for NLP-oriented tasks while preserving interpretability.

Market data. Market completeness is guaranteed by the financial time series source (*yfinance*) during integration. Over 2902 daily buckets, we obtained 5804 price records (two assets: BTC and ETH), with uniform day-level coverage.

6.3 Blind Spots and Temporal Structure

Blind spots (news absence). We define a *blind spot* as a day with valid market data but zero news articles. Let D be the set of integrated daily documents with market prices, and let N_d be the number of news items for day d . The blind spot rate is:

$$\text{BlindSpotRate} = \frac{|\{d \in D : N_d = 0\}|}{|D|} \times 100. \quad (1)$$

In our dataset, 1270 out of 2902 days (43.76%) are blind spots. This highlights a structural sparsity in media coverage that may affect correlation analyses and predictive pipelines.

Continuity of information flow (streaks and gaps). Beyond the blind spot ratio, we characterize the temporal

structure of information availability through consecutive-day runs:

- **News streak:** consecutive days with $N_d \geq 1$.
- **No-news gap:** consecutive days with $N_d = 0$.

The observed mean news streak is 3.26 days, while the mean no-news gap is 2.54 days. These metrics provide an investigative notion of temporal data quality: they quantify whether media attention is persistent (continuous coverage regimes) or fragmented (event-driven bursts separated by silence).

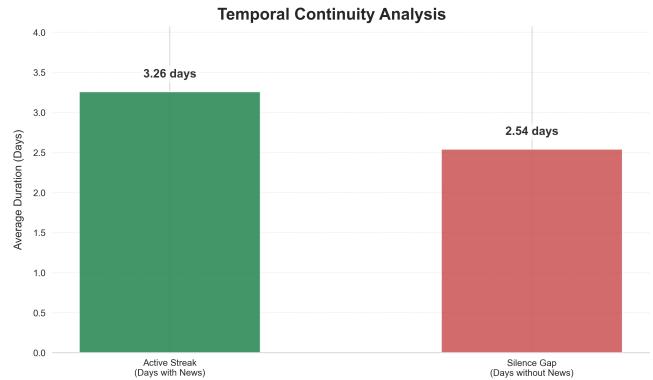


Figure 2: **Temporal Continuity.** Active streaks vs gaps.

6.4 Semantic Redundancy Score

A high article count doesn’t necessarily lead to an increase in information content: multiple articles may echo the same narrative during major events. To quantify it, we compute a **Semantic Redundancy Score** within each day that contains more than one article.

Methodology. For each day d with $M_d > 1$ articles, we represent each article using TF-IDF features computed from *title + abstract*. We then compute pairwise cosine similarity values and average them (excluding self-similarity):

$$\text{Redundancy}_d = \frac{1}{M_d(M_d - 1)} \sum_{i \neq j} \cos(\vec{v}_i, \vec{v}_j). \quad (2)$$

We report:

- the **global redundancy score** as the mean redundancy across all days with $M_d > 1$,
- the set of **high-redundancy days** defined by $\text{Redundancy}_d > 0.20$.

Results and interpretation. The global redundancy score is 2.14%, indicating high *information entropy*: even on multi-article days, the content tends to cover different angles (e.g., regulation, technology, market dynamics) rather than repeating identical narratives. Only 9 days exceed the 0.20 threshold; these outliers can be interpreted as days in which a single shock dominates the media agenda.

Link to EDA (outlier inspection). To investigate whether redundant coverage corresponds to market stress, we further inspected the subset of high-redundancy days. In particular, among the high-redundancy set, we selected the 3 days with the highest article volume and analyzed their dominant themes via word-cloud visualizations and compared their market moves (returns/volatility) against typical days.

Table 6: **High Redundancy Events** (> 0.20). Correlation between narrative convergence and market volatility.

Date	Score	Arts	BTC Vol(\$)	ETH Vol(\$)
2024-05-13	0.4914	9	+1,450.23	+20.55
2022-03-18	0.3499	7	+856.32	+130.91
2019-06-18	0.2017	3	-253.70	-9.27



Figure 3: **Temporal Continuity.** Active streaks vs gaps.

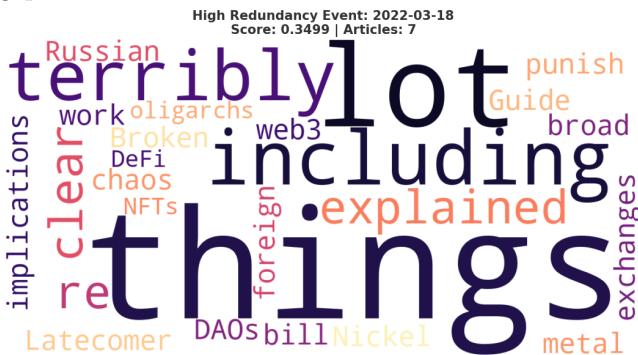


Figure 4: **Temporal Continuity.** Active streaks vs gaps.

Data quality final metrics

- Articles: duplication rate $\approx 0.00\%$; Completeness (missing abstracts) = 0.00% .
- Daily bucket: total days = 2902; blind spots = 1270 days (43.76%); average streak news = 3.26 days, average streak no news = 2.54 days.
- Semantic redundancy score: Average redundancy rate days with news higher than one = 2.14%.

7 Exploratory Data Analysis

The exploratory analysis investigates the relationship between media attention and cryptocurrency market dynamics. This section moves from a macroscopic view of temporal distribution to a granular analysis of semantic content and sentiment during specific market events.

7.1 Temporal Distribution of Media Coverage

We first examined the evolution of the dataset size over time to understand the mainstream media's adoption curve regarding crypto topics. As illustrated in Figure 5,

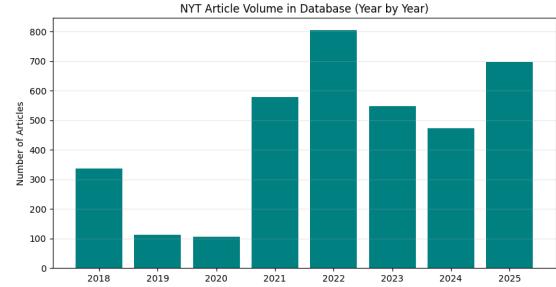


Figure 5: **NYT Article Volume (Year by Year).**

the dataset exhibits a distinct "regime change" in 2021. Between 2018 and 2020 (the so-called "Crypto Winter"), coverage was sparse, averaging fewer than 150 articles per year. A structural break occurred in 2021, where volume nearly tripled, coinciding with the market's all-time highs. Interestingly, the peak volume was not recorded during the price discovery phase but during the crisis year of 2022 (800+ articles). This suggests that mainstream media attention is heavily event-driven, showing a higher responsiveness to market failures than to technological stagnation.

7.2 News Volume vs. Market Trends

To answer **RQ1**, we overlaid the monthly article count against the daily price action of the two largest assets: Bitcoin (BTC) and Ethereum (ETH). Visual inspection of Figures 6 and 7 reveals a high degree of correlation between volatility and media coverage:

- **Volatility Proxy:** The highest volume spikes systematically align with periods of extreme price stress. For instance, the record spike in late 2022 corresponds precisely to the liquidity crisis where Bitcoin dropped below \$16k.
- **Asymmetry:** While bull markets attract consistent coverage, sudden crashes generate disproportionate spikes. This indicates that the media often acts as a "lagging" or "coincident" indicator for distress.

7.3 Semantic Polarity: Positive vs. Negative Themes

Beyond volume, we analyzed the semantic orientation of the coverage using a transformer approach (**FinBERT**) to extract relevant textual features. By categorizing articles based on sentiment scores, we extracted the most frequent terms associated with positive and negative reporting. Figure 8 highlights a stark narrative dichotomy:

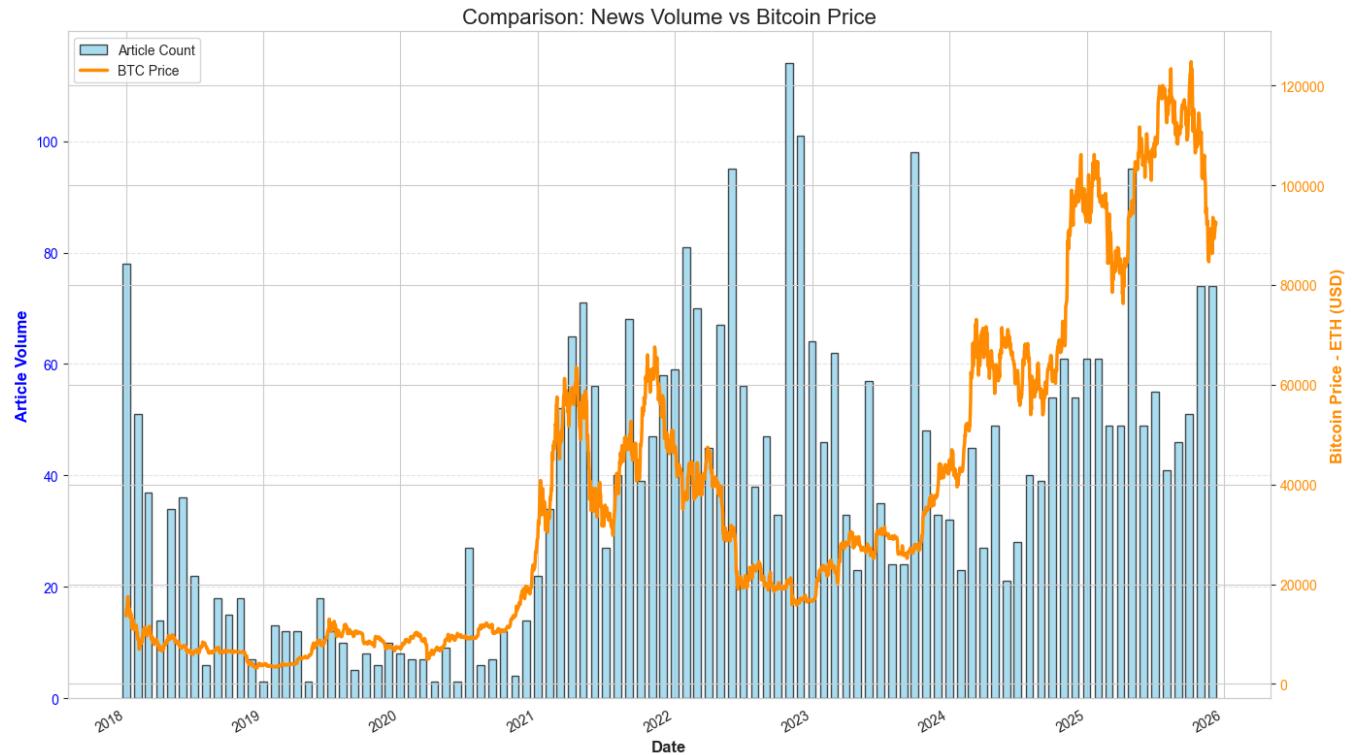


Figure 6: Bitcoin Price vs. News Volume.

- Positive Themes:** The vocabulary is focused on growth and corporate adoption, with terms like "*Tech*", "*Rally*", and "*Investment*".
- Negative Themes:** The negative cloud is significantly more specific and dominated by legal terminology. Keywords such as "*Fraud*", "*Federal*", "*Prosecutor*", and "*Prison*" are prominent, confirming that negative sentiment is largely driven by criminal events rather than pure market performance.

7.4 Impact Analysis: Case Studies

To evaluate the "News Impact" on price (**RQ2**), we isolated two historical highvolatility windows, combining price action, average daily sentiment, and topic modeling.

Case Study 1: The COVID-19 Crash (March 2020)

In March 2020, Bitcoin lost over 50% of its value in two days. The analysis in Figure 9 shows a sentiment score plummeting to **-0.469**. The word cloud captures the external nature of this shock, dominated by "*Coronavirus*", "*Crisis*", and "*Global*". Here, the crypto market behaved as a correlated risk asset.

Case Study 2: China Ban & Elon Musk (May 2021)

In contrast, the May 2021 correction was driven by industry-specific news. Figure 10 shows high volatility triggered by mixed sentiment signals. The word cloud successfully identifies the specific catalysts: "*Elon Musk*", "*Tesla*", and "*China*". This demonstrates the pipeline's

ability to extract specific entities responsible for market movements.

7.5 Keyword Analysis: The "Trump" Factor

Finally, given the political weight of the dataset, we analyzed the narrative surrounding the keyword "President Trump" to understand the intersection of politics and crypto. The analysis in Figure 11 reveals that coverage linking Donald Trump to cryptocurrency is highly geopolitical. Dominant terms include "*Tariff*", "*Trade*", and "*Election*", suggesting that crypto is often framed within broader discussions of US economic policy rather than specific blockchain regulation.

8 Conclusions and Future Work

8.1 Summary of Contributions

This project successfully implemented a robust NoSQL data pipeline capable of ingesting and correlating heterogeneous data sources: unstructured news metadata from the New York Times and structured financial time-series for Bitcoin and Ethereum. By leveraging a decoupled architecture based on Apache Kafka and MongoDB, the system ensured fault tolerance and efficient handling of API rate limits.

The design of the *Daily Market Summary* document, which embeds news articles directly alongside market context and look-ahead price trends ($t+1, t+7$), proved effective for analytical querying. This denormalized schema allowed for the seamless execution of complex queries, such as identifying "information vacuums" and measuring market recovery types ("V-Shape" vs. "U-Shape") without expensive join operations.

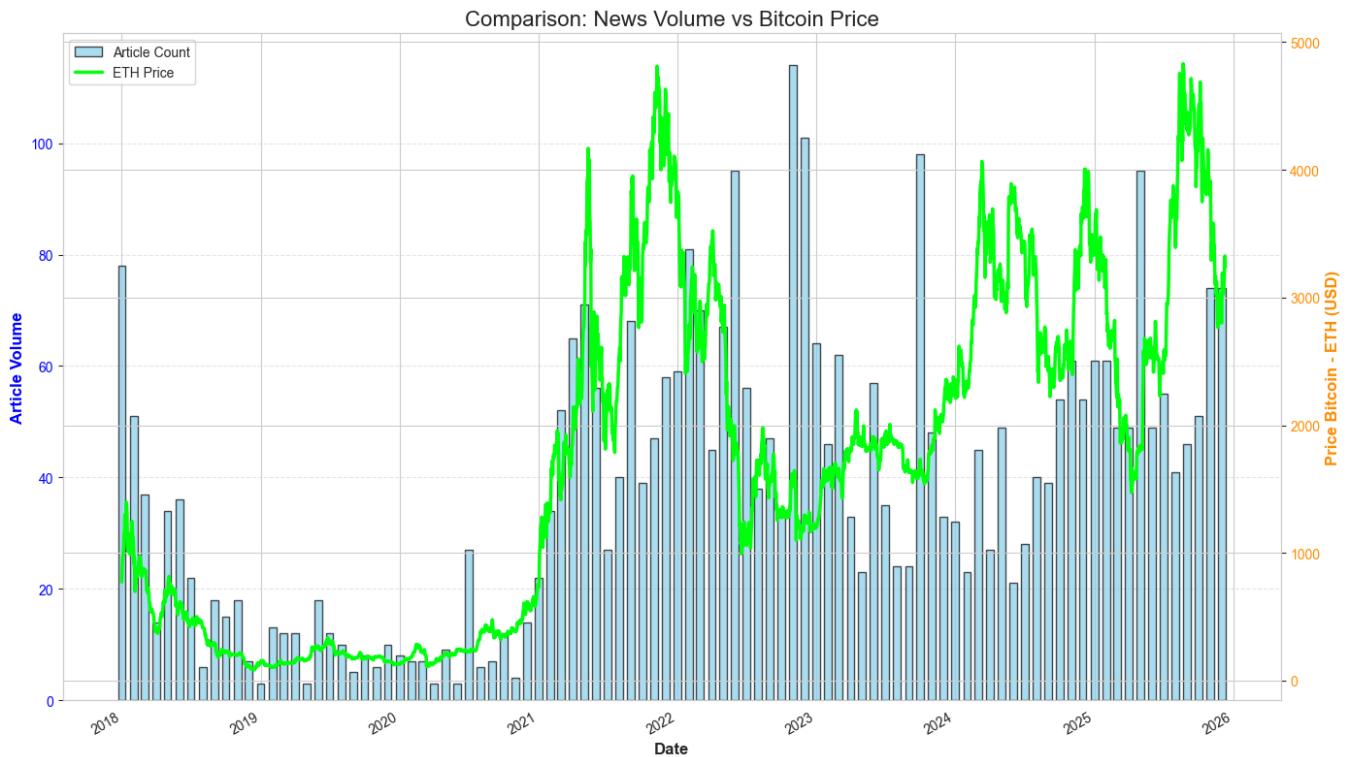


Figure 7: Ethereum Price vs. News Volume.

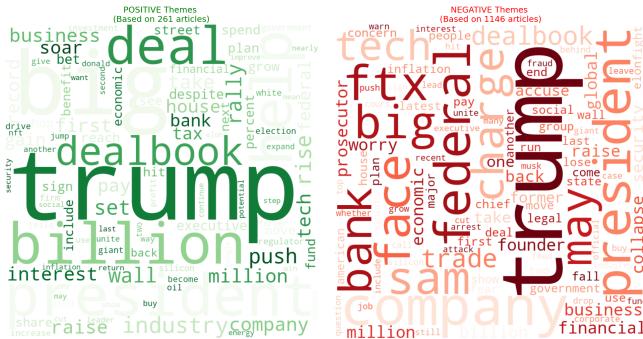


Figure 8: Semantic Polarity Analysis.



Figure 9: Impact Analysis: COVID-19 Crash.

8.2 Key Findings

The Exploratory Data Analysis (EDA) and Data Quality Assessment yielded several critical insights regarding the relationship between mainstream media and crypto markets:

- Volatility Correlation:** News volume is not merely a proxy for interest but a strong indicator of market stress. The highest spikes in coverage systematically

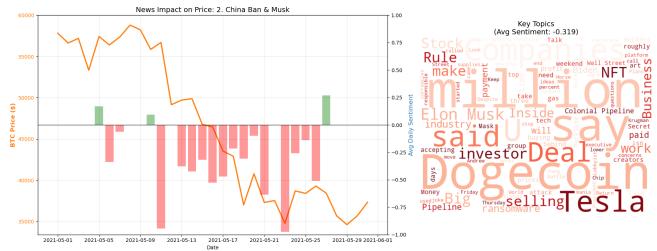


Figure 10: Impact Analysis: China Ban & Elon Musk.

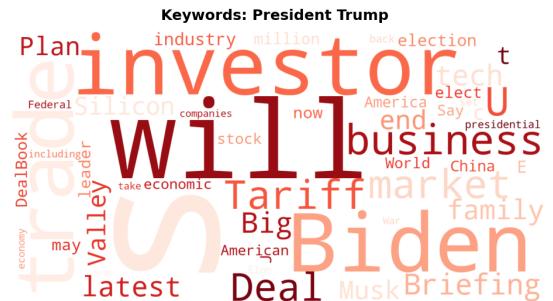


Figure 11: Keyword Focus: 'President Trump'.

aligned with periods of extreme volatility and market failures, such as the FTX collapse in late 2022, suggesting that mainstream media often acts as a lagging or coincident indicator rather than a leading one.

- Sentiment Dichotomy:** Semantic analysis revealed a stark contrast in narrative framing. Positive coverage is generally associated with "Growth" and "Technology," while negative coverage is dominated by "Legal"

and "Fraud" terminology (e.g., "Prosecutor," "Prison"), indicating that negative sentiment is driven more by criminal events than by asset performance.

- **Data Sparsity and Quality:** The quality assessment highlighted a structural "Blind Spot" rate of 43.76%, representing days where market data exists without corresponding NYT coverage. However, the low semantic redundancy score (2.14%) suggests that when news is present, the information entropy is high, providing distinct and non-repetitive narratives.

8.3 Future Work

While the current pipeline establishes a solid foundation for correlation analysis, the findings indicate that volume alone is insufficient to fully explain market behaviors. Future development might focus on three key areas:

1. **Predictive Modeling:** Leveraging the "look-ahead" features already computed in the integration phase, the research will expand to time-series modeling. Techniques such as VAR (Vector Autoregression) or ARIMAX could be trained to test if news sentiment helps forecast next-day volatility.
2. **Source Diversification:** To address the identified "Blind Spots" and the "Event-Driven" bias of mainstream media, the ingestion layer should be augmented with high-frequency social data (e.g., X/Twitter) or crypto-native outlets. This would allow for a comparative analysis of source bias and information latency.

Appendix: Code Listings

Listing 1: First Query

```

1 pipeline_1 = [
2     # 1. Sort for article_count (desc)
3     { "$sort": { "article_count": -1 } },
4
5     # 2. First 5 days
6     { "$limit": 5 },
7
8     # 3. Select what we want to show
9     {
10        "$project": {
11            "date": "$date_str",
12            "#articles": "$article_count",
13            "title": "$articles.title",
14
15            # Bitcoin Data
16            "btc_open": "$prices.BTC-USD.open",
17            "btc_close": "$prices.BTC-USD.close",
18            "btc_volatility_intraday": "$prices.BTC-USD.intraday_change",
19            "btc_volatility_next_week": "$prices.BTC-USD.future_trend.next_week_diff",
20
21            # Ethereum Data
22            "eth_open": "$prices.ETH-USD.open",
23            "eth_close": "$prices.ETH-USD.close",
24            "eth_volatility_intraday": "$prices.ETH-USD.intraday_change",
25            "eth_volatility_next_week": "$prices.ETH-USD.future_trend.next_week_diff"
26        }
27    }
28 ]

```

Listing 2: Second Query

```

1 pipeline_2 = [
2     {
3         # Focus on days where BTC dropped significantly (Intraday < -50)
4         "$match": { "prices.BTC-USD.intraday_change": { "$lt": -50 } }
5     },
6     {
7         "$project": {
8             "Date": "$date_str",
9             "Articles": "$article_count",
10            # --- Bitcoin Analysis ---
11            "BTC_Crash_Amount": { "$subtract": [ "$prices.BTC-USD.close", "$prices.BTC-USD.context.
close_prev_week" ] },
12                "BTC_Immediate_Recovery_24h": "$prices.BTC-USD.future_trend.next_day_diff",
13                "BTC_Weekly_Recovery_7d": "$prices.BTC-USD.future_trend.next_week_diff",
14                "BTC_Recovery_Type": {
15                    "$switch": {
16                        "branches": [
17                            { "case": { "$gt": [ "$prices.BTC-USD.future_trend.next_day_diff", 0 ] }, "t
then": "V-Shape (Fast)" },
18                            { "case": { "$gt": [ "$prices.BTC-USD.future_trend.next_week_diff", 0 ] }, "t
then": "U-Shape (Slow)" }
19                        ],
20                        "default": "No Recovery"
21                    }
22                },
23            # --- Ethereum Analysis ---
24            "ETH_Crash_Amount": { "$subtract": [ "$prices.ETH-USD.close", "$prices.ETH-USD.context.
close_prev_week" ] },
25                "ETH_Immediate_Recovery_24h": "$prices.ETH-USD.future_trend.next_day_diff",
26                "ETH_Weekly_Recovery_7d": "$prices.ETH-USD.future_trend.next_week_diff",
27                "ETH_Recovery_Type": {
28                    "$switch": {
29                        "branches": [
30                            { "case": { "$gt": [ "$prices.ETH-USD.future_trend.next_day_diff", 0 ] }, "t
then": "V-Shape (Fast)" },
31                            { "case": { "$gt": [ "$prices.ETH-USD.future_trend.next_week_diff", 0 ] }, "t
then": "U-Shape (Slow)" }
32                        ],
33                        "default": "No Recovery"
34                    }
35                },
36            },
37        }
38    },
39    { "$sort": { "BTC_Crash_Amount": 1 } }, # Show biggest BTC crashes first
40    { "$limit": 10 }
41 ]

```

Listing 3: Third Query

```

1 pipeline_3 = [
2     # 1. Day with articles and where the price (bitcoin) decrease (next day)
3     {
4         "$match": {
5             "article_count": { "$gt": 0 },
6             "prices.BTC-USD.future_trend.next_day_diff": { "$lt": 0 }
7         }
8     },
9     # 2. Sort by the biggest Bitcoin drop (ascending because values are negative)
10    { "$sort": { "prices.BTC-USD.future_trend.next_day_diff": 1 } },
11    { "$limit": 5 },
12    {
13        "$project": {
14            "date_article": "$date_str",
15            "news_count": "$article_count",
16            "title": { "$arrayElemAt": ["$articles.title", 0] }, # First article title
17
18            # Bitcoin Data
19            "btc_price_close": "$prices.BTC-USD.close",
20            "btc_loss_next_day": "$prices.BTC-USD.future_trend.next_day_diff",
21            "btc_next_week_diff": "$prices.BTC-USD.future_trend.next_week_diff",
22
23        }
24    }
25 ]

```

Listing 4: Fourth Query

```

1 pipeline_4 = [
2     {
3         "$group": {
4             "_id": {
5                 # Conditional logic to label groups
6                 "$cond": [{ "$gt": ["$article_count", 0] }, "Days WITH News", "Days WITHOUT News"]
7             },
8             "total_days": { "$sum": 1 },
9
10            # Calculate absolute variation mean for Bitcoin
11            "btc_avg_abs_volatility": { "$avg": { "$abs": "$prices.BTC-USD.intraday_change" } },
12
13            # Calculate absolute variation mean for Ethereum
14            "eth_avg_abs_volatility": { "$avg": { "$abs": "$prices.ETH-USD.intraday_change" } }
15        }
16    },
17    {
18        "$project": {
19            "_id": 0, # Hide the default MongoDB ID field
20            "Category": "$_id", # Move the label to a column named 'Category',
21            "total_days": 1, # Keep this column
22            "btc_avg_abs_volatility": 1, # Bitcoin Volatility
23            "eth_avg_abs_volatility": 1 # Ethereum Volatility
24        }
25    }
26 ]

```

Listing 5: Fifth Query - Part 1

```

1 pipeline_5 = [
2
3     # 1. Blind spot: zero articles + BTC move
4     {
5         "$match": {
6             "article_count": 0,
7             "prices.BTC-USD.intraday_change": { "$exists": True }
8         }
9     },
10
11    # 2. Volatility + Time windows (3 days before/after)
12    {
13        "$addFields": {
14            "btc_abs_volatility": {
15                "$abs": "$prices.BTC-USD.intraday_change"
16            },
17            "start_3d_before": {
18                "$subtract": ["$date", 3 * 24 * 60 * 60 * 1000]
19            },
20            "end_3d_after": {
21                "$add": ["$date", 3 * 24 * 60 * 60 * 1000]
22            }
23        }
24    },
25
26    # 3. Lookup: 3 days BEFORE
27    {
28        "$lookup": {
29            "from": "daily_market_summary",
30            "let": { "start": "$start_3d_before", "blind": "$date" },
31            "pipeline": [
32                { "$match": { "$expr": { "$and": [
33                    { "$gte": ["$date", "$$start"] },
34                    { "$lt": ["$date", "$$blind"] }
35                ] } } },
36                { "$group": { "_id": null, "total_articles": { "$sum": "$article_count" } } }
37            ],
38            "as": "days_before"
39        }
40    },

```

Listing 6: Fifth Query - Part 2 (continued)

```

45 # 4. Lookup: 3 days AFTER
46 {
47     "$lookup": {
48         "from": "daily_market_summary",
49         "let": { "blind": "$date", "end": "$end_3d_after" },
50         "pipeline": [
51             { "$match": { "$expr": { "$and": [
52                 { "$gt": ["$date", "$$blind"] },
53                 { "$lte": ["$date", "$$end"] }
54             ] } } },
55             { "$group": { "_id": null, "total_articles": { "$sum": "$article_count" } } }
56         ],
57         "as": "days_after"
58     },
59 },
60
61 # 5. Flatten lookup results
62 {
63     "$addFields": {
64         "articles_3d_pre": { "$ifNull": [{ "$arrayElemAt": ["$days_before.total_articles", 0]
65 }, 0] },
66         "articles_3d_post": { "$ifNull": [{ "$arrayElemAt": ["$days_after.total_articles", 0]
67 }, 0] }
68     },
69
70 # 6. Sort by risk (volatility) and Final Output
71 { "$sort": { "btc_abs_volatility": -1 } },
72 { "$limit": 5 },
73 {
74     "$project": {
75         "_id": 0, "date": "$date_str",
76         "articles_today": "$article_count",
77         "articles_3d_pre": 1, "articles_3d_post": 1,
78         "btc_daily_change": "$prices.BTC-USD.intraday_change",
79         "btc_absolute_volatility": "$btc_abs_volatility",
80         "eth_daily_change": "$prices.ETH-USD.intraday_change"
81     }
82 }

```