

# AINewsQuake

Price Chart with News Annotations

Data Management Project Presentation

Abbas Taqvi    Umeir Mohamed    MohammadAmin Saberi

University of Milano-Bicocca (UNIMIB)

January 10, 2026



# Agenda

- Motivation & Objectives
- Research Questions
- Data Sources & Acquisition
- Storage & Database Design
- Data Quality & Integration
- Dashboard & Analysis
- Limitations & Future Work
- Conclusion

Financial markets react quickly to information. AI-related headlines can trigger significant short-term price movements, volatility spikes, and abnormal trading volumes.

**Goal:** Build a reproducible data management pipeline that integrates **AI news sentiment** with **high-frequency market data** and enables interactive exploration of their relationship.

# Objectives

- Acquire heterogeneous data: **news events** + **1-min OHLCV**.
- Enrich news with **sentiment scores** (VADER compound).
- Store large time-series data efficiently using **TimescaleDB**.
- Provide reproducible ETL with **idempotent loads** and validation.
- Enable analysis via **SQL queries** and a **Streamlit dashboard**.

# Research Questions

## RQ1

How does AI-related news sentiment correlate with intraday volatility and price movement?

## RQ2

Do certain AI-centric stocks react more strongly to positive/negative headlines?

## RQ3

How complete and reliable is the integrated dataset (quality + integration coverage)?

## **Finnhub (News API)**

- Historical company news (2025)
- Headline, timestamp, source
- Rate limits + pagination (250 items/request)

## **Databento (Market Data)**

- 1-minute OHLCV bars (2025)
- High reliability for intraday bars
- Large data volume → time-series DB needed

- **Backfill strategy:** fetch news backwards due to Finnhub limits.
- **Validation:** Pydantic schemas ensure type correctness.
- **Repeatable runs:** ETL is designed to be idempotent.

- PostgreSQL + time-series extension.
- **Hypertables** partition tick data by time.
- Efficient range queries + indexing on (`time`, `ticker`).
- **Compression policy** for older data (e.g.,  $< 7$  days).

**Why this matters:** 1-min OHLCV for 10 tickers across a year  $\rightarrow$  millions of rows.



# Database Schema (Logical)

- `ai_news_events` (relational table)
  - `event_id` (PK), `ticker`, `published_at`, `headline`, `source`
  - `sentiment_score`  $\in [-1, 1]$
- `market_ticks` (TimescaleDB hypertable)
  - primary key: (`time`, `ticker`)
  - `open`, `high`, `low`, `close`, `volume`

- **Completeness:** missing fields (timestamps, OHLCV, headline).
- **Validity:** sentiment score range, non-negative volume.
- **Uniqueness:** unique event IDs; unique (time, ticker).
- **Consistency:** timezone normalization (UTC) across sources.

# Idempotent ETL & De-duplication

- News insert: `ON CONFLICT DO NOTHING`
- Market insert: `ON CONFLICT DO UPDATE`
- Ensures reproducibility and safe re-runs.
- Prevents duplicate rows and supports partial backfills.

- Challenge: align **irregular news timestamps** with **regular 1-min candles**.
- Keep sources separate and integrate at query-time.
- Supports flexible alignment windows (nearest candle / pre-post event windows).

- **Coverage:** fraction of news events mapped to a market bar.
- **Latency:** time difference between news and nearest candle.
- **Ambiguity:** multiple news events mapped to the same candle.

- Interactive candlestick chart with sentiment-coded news markers.
- Hover tooltips show headline, sentiment score, and time.
- Filters: ticker selection + date range.
- Summary metrics: number of events, average sentiment, volume, price change.

**Demo:** *Price Chart with News Annotations*

- Identify volatility clusters around major AI headlines.
- Compare average sentiment vs returns across tickers.
- Detect extreme price moves and inspect associated news.

- Docker Compose for TimescaleDB deployment.
- uv for dependency management.
- .env configuration for API keys.
- One-command ETL runs + Streamlit launch.

**Repository:** [github.com/smabbasht/ainewsquake](https://github.com/smabbasht/ainewsquake)



- Free-tier API limits can affect backfill completeness.
- VADER sentiment is generic; domain-specific models may improve quality.
- Correlation  $\neq$  causation; causal inference is future work.

- Add anomaly detection to identify “quake” events systematically.
- Build automated DQ dashboards (metrics over time).
- Extend to more assets (crypto/forex) and additional APIs.
- Improve temporal integration with richer pre/post windows.

- End-to-end pipeline: acquisition → storage → profiling → integration → analysis.
- Efficient time-series database design using TimescaleDB.
- Robust ETL with validation + idempotent loads.
- Interactive dashboard for exploring sentiment-volatility relationships.

**Thank you!**

Questions?