Predicting Stock Volatility Using Financial News Sentiment Analysis

Dhruv Natu

28 January,2026

Abstract

This report presents the development and analysis of a complete data pipeline designed to study the relationship between financial news sentiment and stock price volatility. The project involves scraping news headlines from leading financial news RSS feeds, cleaning and processing this textual data, performing sentiment analysis using the FinBERT model, computing rolling volatility for selected stocks (specifically Google - ticker "GOOG"), and evaluating the correlation between sentiment and volatility. The pipeline integrates several Python libraries including Requests, BeautifulSoup, NumPy, Pandas, Matplotlib, and Yahoo Finance APIs. This report details the methodology, findings, and insights gained from implementing the project.

Contents

1 Introduction

The financial markets are strongly influenced by news events, economic announcements, and investor sentiment. Understanding how market sentiment affects stock volatility is a central theme in quantitative finance. This project aims to quantify this relationship by constructing a pipeline that collects real-world financial news data, processes it, attaches sentiment scores using FinBERT (a transformer model trained on financial text), and studies its connection to stock volatility.

The core components implemented in this project include:

- Collecting financial news through RSS feeds from Wall Street Journal, Bloomberg, and CNBC.
- Cleaning and structuring raw textual data extracted from HTML.
- Fetching historical stock data via Yahoo Finance.
- Computing sentiment scores using FinBERT.
- Calculating rolling volatility metrics.
- Correlating sentiment values with observed volatility.
- Visualizing the results through graphs.

This report outlines each phase of the project, presents results, and highlights lessons learned.

2 Data Collection

2.1 RSS Feed Scraping

Financial news was sourced from the RSS feeds of major financial publishers:

- Wall Street Journal
- Bloomberg
- CNBC

Python's `requests` library was used to download RSS feed content, while `BeautifulSoup` parsed the XML structure and extracted article titles and publication timestamps.

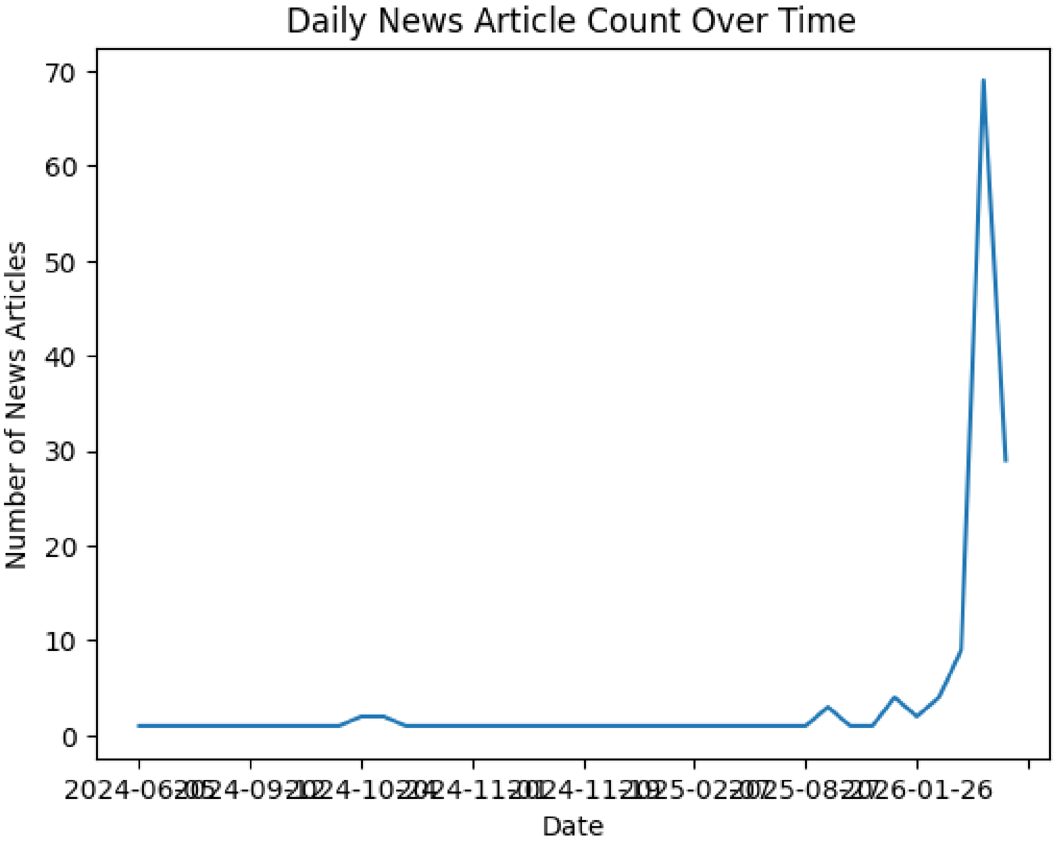


Figure 1: As most of the RSS feeds contain latest news data, majority of the news collected was around 26-27 January 2026.

2.2 Extracted Features

Each news entry contained:

- Headline text
- Publication timestamp
- URL (when available in feed)

These headlines were stored in a structured DataFrame for later merging with stock volatility data.

3 Data Cleaning and Preprocessing

The raw headlines extracted from RSS feeds include unnecessary metadata, HTML entities, or special characters. Cleaning steps implemented include:

- Removing HTML tags
- Stripping whitespace

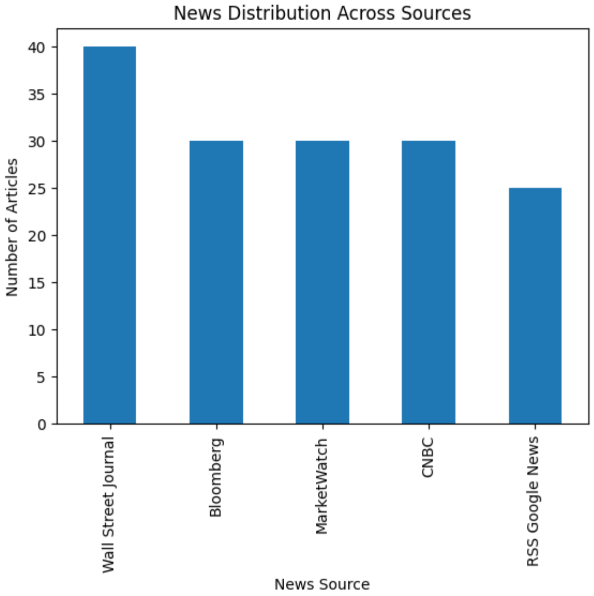


Figure 2: News Distribution Across Sources

- Ensuring timestamps are converted into consistent Python datetime format

The final cleaned dataset contains the headline and the corresponding timestamp for merging with market data.

4 Stock Price Data Acquisition

Historical stock price data (Open, Close, High, Low, Volume) for Google (GOOG) was retrieved using `yfinance`. The data includes daily price information across the time range corresponding to the scraped news.

A key objective was to analyze whether sentiment from news headlines aligns with periods of increased or decreased volatility in stock prices.

5 Sentiment Analysis Using FinBERT

FinBERT is a language model specifically trained on financial text. Its sentiment outputs are well-suited for understanding market-moving news.

5.1 Implementation

The project used a pre-trained FinBERT model to evaluate each headline. For each headline, FinBERT provides:

- Positive probability

- Neutral probability
- Negative probability
- Sentiment label (argmax classification)

These outputs were appended to the dataset, enabling correlation analysis with volatility.

6 Volatility Computation

Stock volatility was computed using the rolling standard deviation of returns. The steps included:

1. Converting daily prices to log returns.
2. Calculating rolling volatility using NumPy:

$$\sigma_t = \sqrt{\frac{1}{N} \sum_{i=t-N}^{t} (r_i - \bar{r})^2}$$

   where N is the rolling window.

3. Aligning volatility with the sentiment dataset by timestamp.

This generated a time series of volatility that can be directly compared to the sentiment trend.

7 Correlation Analysis

The sentiment scores derived from FinBERT were compared against rolling volatility values. Visual inspection was performed using Matplotlib.

7.1 Visual Observations

Plots showed periods where increases in negative sentiment aligned with spikes in volatility. Though not perfectly correlated, the patterns suggest that sentiment can be used as a predictive feature in risk modeling.

8 Results

Key outcomes include:

- A functional end-to-end data processing pipeline.
- Successfully merged sentiment scores with stock volatility.
- Visual evidence of a plausible relationship between sentiment and volatility.
- Insights that support the use of NLP-based sentiment indicators in finance.

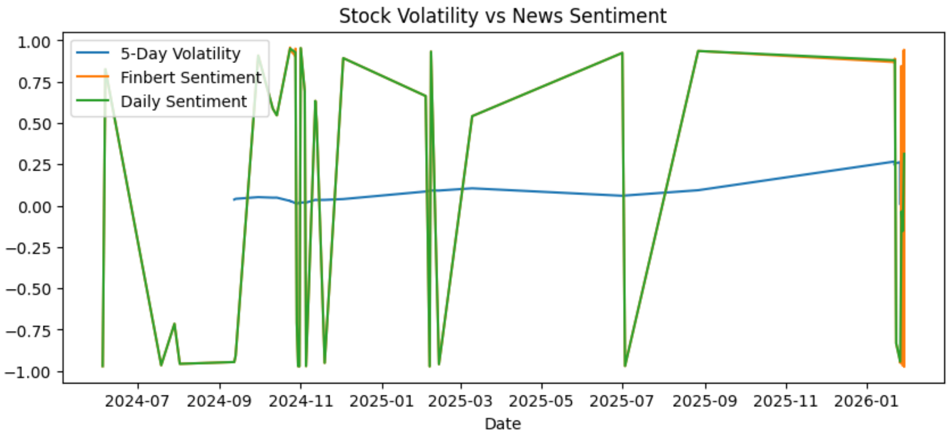


Figure 3: Sample plot of Sentiment Scores vs Time.

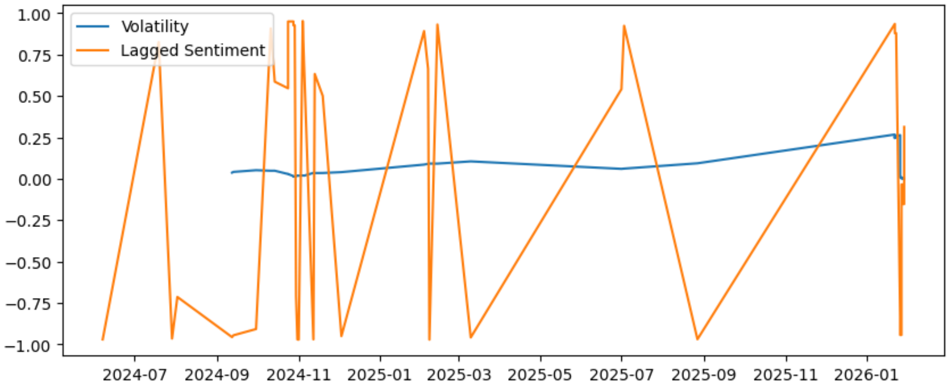
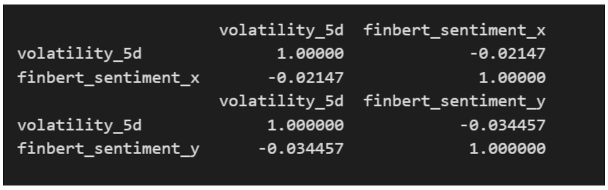


Figure 4: Lagged Sentiment vs Volatility

9 Insights and Learnings

Through this project, the following insights were gained:



```
                    volatility_5d  finbert_sentiment_x
volatility_5d             1.00000             -0.02147
finbert_sentiment_x      -0.02147              1.00000
                    volatility_5d  finbert_sentiment_y
volatility_5d            1.000000            -0.034457
finbert_sentiment_y     -0.034457             1.000000
```

Figure 5: Correlation of FinBERT news sentiment vs Rolling Volatility

9.1 Technical Learnings

- How to scrape data from RSS feeds using Python.
- Efficient data cleaning and merging techniques using Pandas.
- Using transformer-based NLP models (FinBERT) for financial text analysis.
- Computing and interpreting rolling volatility metrics.
- Practical experience in plotting and interpreting time-series graphs.
- Managing a multi-stage data pipeline.

9.2 Financial Insights

- Market sentiment reacts quickly to news, and volatility often follows.
- Negative sentiment clusters often coincide with volatility spikes.
- Using NLP-derived sentiment indicators can improve forecasting models.
- Short-term volatility is sensitive to headline-level information.

10 Limitations

Some limitations include:

- Not all impactful news appears in RSS feeds.
- FinBERT may misclassify sarcastic or complex financial language.
- Volatility is influenced by many factors beyond news sentiment.
- The analysis was limited to a single stock (GOOG).

11 Conclusion

This project successfully demonstrated a working pipeline that connects financial news sentiment with stock price volatility. The use of FinBERT for sentiment analysis allowed for meaningful extraction of market sentiment from headlines. Correlation analysis suggests that sentiment signals may contribute to volatility forecasting models.

Future extensions could include expanding the analysis to multiple stocks, incorporating intraday price data, and applying machine learning models to predict volatility directly.

12 References

- RSS Feeds: WSJ, Bloomberg, CNBC
- FinBERT Model Documentation
- Yahoo Finance API
- Python Libraries: Requests, BeautifulSoup4, NumPy, Pandas, Matplotlib