# Market Mood and Moves

A Sentiment-Driven Approach to Financial Markets

**Pranit Anand**

Roll Number: *24B3914*

WiDS 5.0 – Analytics Club

January 2026

# Contents

# 1 Week 1: Foundations of Sentiment-Driven Financial Systems

## 1.1 Project Motivation and Initial Understanding

During the first week of the *Market Mood & Moves* project, the primary focus was on developing a conceptual understanding of how sentiment-driven financial systems are structured. Rather than immediately working on predictive models, the emphasis was placed on understanding the motivation behind using sentiment as an explanatory signal for short-term market movements.

Through the study of behavioral finance, it became clear that classical financial models often fail to account for psychological and social factors that influence investor behavior. This week marked a shift in perspective, where markets were no longer viewed as purely rational systems, but as environments shaped by perception, narrative, and information flow.

An important learning outcome was understanding the complete system architecture at a high level. The pipeline introduced during this week integrates financial price data, textual news data, and machine learning models. Gaining clarity on this architecture before implementation helped establish context for all subsequent experimentation.

## 1.2 Learning Behavioral Finance Concepts

A significant portion of Week 1 was dedicated to understanding the principles of behavioral finance and how they differ from traditional financial theory. Concepts such as herd behavior, loss aversion, and overconfidence were studied not as abstract ideas, but as practical explanations for observed market anomalies.

The notion of herd behavior was particularly insightful, as it provided a theoretical justification for why news events can trigger synchronized market reactions. This reinforced the idea that sentiment extracted from news can serve as a proxy for collective investor psychology rather than a measure of intrinsic value.

Through these discussions, sentiment analysis was reframed as a tool for capturing short-term market mood rather than predicting long-term fundamentals.

## 1.3 Understanding the End-to-End System Workflow

Week 1 placed strong emphasis on understanding the complete workflow of the system before implementing individual components. The architecture was broken down into logical layers, each with a distinct responsibility:

1. Data ingestion using APIs

2. Storage of raw data in structured formats
3. Sentiment extraction using NLP models
4. Temporal alignment to prevent data leakage
5. Preparation of features for downstream modeling

Analyzing this workflow helped clarify why preprocessing steps such as alignment and aggregation are not optional but foundational. One important realization was that errors introduced at early stages can silently propagate through the pipeline and invalidate later results, regardless of model sophistication.

## 1.4 Hands-On Learning with Python, Pandas, and NumPy

Week 1 also reinforced practical skills using Python and core data science libraries. Through hands-on experimentation, Pandas was used to load, explore, and manipulate tabular datasets containing news and stock price information.

Common operations such as filtering rows, computing daily returns, and aggregating values by date were explored. NumPy supported numerical operations required for financial time series, including differencing and basic statistical computations.

A key learning from this phase was recognizing how subtle implementation issues — such as incorrect indexing, missing values, or implicit type conversions — can introduce silent errors into financial datasets. This reinforced the need for careful validation at every stage of data processing.

## 1.5 Initial Exposure to NLP Preprocessing

Week 1 introduced the foundational NLP preprocessing steps required to handle financial text. Tokenization, stop-word removal, and lemmatization were studied and applied as part of exploratory experimentation.

While these steps are standard in NLP pipelines, experimentation revealed that over-aggressive preprocessing can remove financially meaningful terms. This highlighted the importance of balancing noise reduction with semantic preservation, especially in domains where subtle wording changes can alter sentiment interpretation.

## 1.6 Comparing Sentiment Analysis Approaches

Two sentiment analysis techniques were explored conceptually during this week: VADER and FinBERT. VADER served as a lightweight baseline for quick sentiment estimation, while FinBERT illustrated the advantages of domain-specific language models.

Studying these approaches introduced the concept of domain shift, helping explain why general-purpose sentiment models often misinterpret financial language. This understanding directly influenced the choice of sentiment models in later experimentation.

### 1.7 Learning API-Based Data Collection

A practical learning outcome of Week 1 was understanding how financial and news data are collected using APIs. Experimentation with NewsAPI and `yfinance` demonstrated how real-world data ingestion differs from curated academic datasets.

Several limitations were encountered during this process, including rate limits, restricted historical access, and the need for secure API key handling. These constraints highlighted the gap between idealized data pipelines and production-oriented systems.

### 1.8 Exposure to Industry-Oriented Challenges

Week 1 also introduced several challenges commonly faced in industry applications. Experiments revealed that CSV-based storage is often insufficient for scalable systems, motivating the use of databases and columnar formats.

Keyword-based news retrieval exposed the problem of semantic ambiguity, where company names can refer to unrelated entities. Additionally, the non-stationary nature of raw stock prices and the importance of correct timezone alignment were identified as critical issues.

These challenges reinforced the idea that correctness, causality, and data integrity are as important as modeling techniques in financial analytics.

### 1.9 Reflections and Key Learnings from Week 1

By the end of Week 1, the focus had shifted from model-centric thinking to a system-oriented mindset. The week emphasized that sentiment-driven trading systems are interdisciplinary in nature, requiring careful coordination between finance theory, NLP, data engineering, and statistical reasoning.

This foundational understanding informed all subsequent experimentation and helped establish realistic expectations for later modeling stages.

## 2 Week 2: Financial NLP and Domain-Specific Sentiment Modeling

Week 2 focused on developing a deeper understanding of Natural Language Processing (NLP) techniques as applied to financial text. Building on the behavioral finance perspective introduced earlier, this week explored how unstructured news data can be transformed into structured sentiment signals suitable for quantitative analysis.

The emphasis during this phase was on understanding the principles behind modern language models rather than on extensive model training. This helped establish a clear conceptual foundation for how sentiment extraction fits into the overall project pipeline.

## 2.1 Understanding the Limitations of Classical Sentiment Methods

The week began with a review of traditional sentiment analysis approaches such as lexicon-based and rule-driven methods. These techniques assign fixed sentiment scores to words and aggregate them to obtain an overall sentiment score.

Through study and examples, it became clear that such approaches are often insufficient for financial text. Many commonly used financial terms have meanings that depend heavily on context. Words such as “liability” or “volatile” may describe neutral financial concepts rather than negative sentiment. This observation highlighted why static sentiment dictionaries can lead to misleading interpretations when applied directly to financial news.

## 2.2 From Static to Contextual Word Representations

Week 2 introduced the evolution of word representations in NLP. Early embedding models such as Word2Vec and GloVe represent each word using a single fixed vector. While effective for capturing general semantic similarity, these models struggle with polysemy, where a word has multiple meanings depending on context.

In financial language, this limitation is especially pronounced. Understanding this issue provided motivation for contextual embeddings, which generate word representations that adapt based on surrounding text rather than relying on isolated tokens.

## 2.3 Learning the Transformer Architecture

The Transformer architecture was studied as the foundation of modern NLP models. A key learning was understanding how self-attention allows each word in a sentence to incorporate information from all other words simultaneously.

This mechanism is particularly relevant for financial headlines, where sentiment can be influenced by qualifiers, numerical values, or negations. The ability of Transformers to process text in parallel was also noted as an important factor in their scalability to large text corpora.

## 2.4 BERT and Bidirectional Context

BERT was examined as a representative Transformer-based language model. The concept of bidirectional context was a central learning outcome, as it allows the model to consider both preceding and following words when interpreting meaning.

The structure of BERT's input representation, including token embeddings, positional embeddings, and segment embeddings, was studied to understand how raw text is encoded numerically. WordPiece tokenization was noted as particularly useful for handling rare or compound financial terms.

## 2.5 Pre-Training Objectives in BERT

Week 2 also covered the objectives used to pre-train BERT. Masked Language Modeling (MLM) was studied as a method for learning contextual word relationships, while Next Sentence Prediction (NSP) was introduced as a way to capture inter-sentence coherence.

Understanding these objectives helped clarify why BERT is able to generalize effectively across a wide range of NLP tasks.

## 2.6 Domain Shift and Financial Language

A key conceptual topic introduced during this week was domain shift. General language models are trained on data sources such as books and encyclopedias, which differ significantly from financial news in style and vocabulary.

Recognizing this mismatch helped explain why general-purpose models may not perform optimally on financial sentiment tasks and motivated the use of domain-adapted models.

## 2.7 Understanding FinBERT

FinBERT was studied as a domain-adapted extension of BERT designed specifically for financial sentiment analysis. Learning about FinBERT's additional pre-training on financial corpora provided insight into how domain adaptation improves performance on specialized text.

Rather than focusing on model training, the emphasis was on understanding how FinBERT's design addresses the limitations of general-purpose language models in financial contexts.

## 2.8 Interpreting Sentiment Outputs

An important learning outcome was understanding sentiment classification as a probabilistic process. FinBERT produces probabilities for positive, negative, and neutral sentiment classes, which allows sentiment to be interpreted with an associated level of confidence.

This probabilistic interpretation is particularly useful when aggregating sentiment across multiple news articles, as required in later stages of the project.

### 2.9 Key Learnings from Week 2

By the end of Week 2, a clearer understanding had developed of how modern NLP models represent and interpret financial text. The study of Transformers, BERT, and FinBERT reinforced the importance of domain-specific modeling and careful interpretation of sentiment signals.

These learnings directly informed how sentiment features were constructed and used in subsequent time series modeling and system integration.

## 3 Week 3: Sequence Modeling and Temporal Dynamics in Financial Markets

### 3.1 Motivation: The Role of Time in Financial Data

During the first two weeks of the project, the focus was primarily on building data pipelines and extracting sentiment signals from financial news. Week 3 introduced a critical realization: both stock prices and sentiment are fundamentally temporal. A market observation cannot be interpreted in isolation, as its significance depends heavily on recent historical behavior.

For example, a price level may indicate strength or weakness depending on the preceding trend. This observation motivated the transition from static feature analysis toward sequence-based modeling approaches that explicitly account for temporal dependence.

### 3.2 Time Series as Sequential Data

Financial market data naturally takes the form of a time-ordered sequence:

$$x_1, x_2, \ldots, x_T$$

Rather than assuming independence between observations, time series modeling aims to capture the conditional structure:

$$P(x_t \mid x_1, \ldots, x_{t-1})$$

This probabilistic perspective clarified why classical machine learning models, which rely on the Independent and Identically Distributed (I.I.D.) assumption, are often inadequate for financial prediction tasks. Markets exhibit serial correlation, delayed reactions, and regime persistence, all of which require models that can incorporate historical context.

## 3.3 Introduction to Recurrent Neural Networks

Recurrent Neural Networks (RNNs) were studied as an initial approach to sequence modeling. RNNs maintain a hidden state that is updated at each time step, allowing information from previous observations to influence future outputs:

$$h_t = f(x_t, h_{t-1})$$

This recursive structure enables parameter sharing across time and allows the model to process sequences of arbitrary length. Conceptually, the hidden state acts as a compressed summary of past information.

However, studying RNNs also revealed their limitations. In practice, vanilla RNNs struggle to learn long-range dependencies due to vanishing and exploding gradient problems during backpropagation through time. This limitation is particularly relevant in financial data, where important effects may manifest over extended horizons.

## 3.4 Long Short-Term Memory (LSTM) Networks

Long Short-Term Memory (LSTM) networks were introduced as an extension of RNNs designed to address long-term dependency issues. The key conceptual advancement in LSTMs is the explicit separation of long-term memory and short-term output through a gated architecture.

An LSTM maintains:

- A cell state $C_t$, which carries long-term information
- A hidden state $h_t$, which represents the immediate output

Three gates regulate information flow:

- The forget gate determines which historical information is retained
- The input gate controls how new information is incorporated
- The output gate determines what information is exposed to the next layer

The cell update equation,

$$C_t = F_t \odot C_{t-1} + I_t \odot \tilde{C}_t,$$

illustrates how LSTMs balance memory retention with adaptability. Studying this architecture provided insight into why LSTMs are widely used in financial time series modeling.

## 3.5 Multimodal Feature Representation

A central conceptual contribution of Week 3 was learning how heterogeneous data sources can be combined into a unified temporal representation. Instead of modeling prices alone, the framework integrates sentiment information extracted from financial news.

Each time step can be represented by a feature vector containing:

- Price-based features such as returns
- Basic technical indicators capturing recent market behavior
- Aggregated sentiment scores derived from news headlines

These features are organized into a three-dimensional tensor of the form:

$$(\text{Batch Size}, \text{Sequence Length}, \text{Number of Features})$$

This representation reflects the intuition that price data encodes historical outcomes, while sentiment captures market expectations and collective reactions.

## 3.6 Sliding Window Supervision

Week 3 also introduced the sliding window approach used to convert time series data into supervised learning samples. For a window length $L$, the model uses a sequence of $L$ consecutive observations to predict the next time step.

This framing preserves temporal ordering while enabling gradient-based optimization. Understanding correct window construction was emphasized as a conceptual requirement, as misalignment can lead to information leakage or unrealistic predictions.

## 3.7 Observations from Experimentation

Limited experimentation with sequence-based modeling highlighted several practical considerations:

- Financial time series are often noisy and exhibit strong persistence
- Small datasets limit the depth of temporal patterns that can be learned
- Sequence length selection involves a trade-off between context and data availability

These observations reinforced that while LSTMs are powerful, their effectiveness depends heavily on data quality, feature design, and realistic evaluation protocols.

### 3.8 Interpreting Model Outputs

An important takeaway from Week 3 was that sequence models produce numerical predictions, not trading decisions. Predicted values must be interpreted within a broader decision-making framework that accounts for uncertainty, risk, and market conditions.

This understanding set the stage for integrating sentiment signals, price dynamics, and alignment logic into a cohesive system rather than treating model outputs in isolation.

### 3.9 Summary of Week 3 Learnings

Week 3 marked a conceptual shift from static analysis to dynamic modeling. The key learnings include:

- Financial data requires sequence-aware modeling
- Temporal dependencies violate I.I.D. assumptions
- LSTMs provide a structured way to retain historical information
- Sentiment is most useful when combined with price-based features
- Model predictions must be interpreted carefully within realistic decision frameworks

This week established the temporal perspective required for full system integration in the final phase of the project.

## 4 Week 4: End-to-End Integration and System-Level Design

### 4.1 Objective and Perspective

Week 4 focused on understanding how the individual components developed in earlier weeks fit together into a coherent end-to-end system. Rather than optimizing individual models, the emphasis was on studying data flow, temporal correctness, and realistic system constraints in sentiment-driven financial pipelines.

This week marked a shift from algorithm-centric thinking to system-level reasoning.

### 4.2 Modular Pipeline Design

The overall pipeline was conceptually decomposed into independent stages:

1. News ingestion and raw data storage
2. Sentiment extraction using FinBERT

3. Temporal alignment to trading days
4. Aggregation and merging with price data
5. Sequence modeling using LSTM-based methods

This modular structure improves interpretability and makes it easier to isolate errors or inconsistencies at each stage.

## 4.3 News Ingestion and Sentiment Extraction

News articles are collected via APIs and stored in raw form before processing. Sentiment extraction is performed using FinBERT, producing probabilistic scores for positive, negative, and neutral sentiment.

Retaining probabilities rather than hard labels allows sentiment to be treated as a continuous signal suitable for aggregation and time series modeling.

## 4.4 Temporal Alignment and Bias Prevention

Correct time alignment was identified as a critical requirement. News timestamps are mapped to trading days using market hours and weekend rules to prevent look-ahead bias.

This step ensures that the model does not inadvertently learn from future information.

## 4.5 Aggregation and Feature Fusion

Multiple news articles on the same trading day are aggregated using simple statistics such as average sentiment and article count. These features are then merged with historical stock prices to form a unified dataset.

This fusion reflects a multimodal view of markets, combining realized price movements with sentiment-driven expectations.

## 4.6 Sequence Construction and Practical Observations

The merged data is transformed into sequences using a sliding window approach for temporal modeling. A key observation from this phase was data sparsity: strict company-specific filtering often yields very few news articles, limiting the effectiveness of deep sequence models.

This highlighted the trade-off between relevance and data availability in real-world financial systems.

### 4.7 Key Learnings from Week 4

Week 4 reinforced several system-level insights:

- Financial ML systems are heavily constrained by data availability
- Temporal correctness is more important than model complexity
- Sentiment signals are noisy but informative when aggregated
- End-to-end design choices strongly influence model behavior

This week completed the integration of behavioral finance, NLP, and time series modeling into a single analytical framework.

## 5 Challenges and Limitations

Throughout the course of the Market Mood and Moves project, several practical challenges and limitations were encountered. These challenges were not unexpected and closely resemble issues faced in real-world financial analytics and data science workflows.

### 5.1 Data Availability and API Constraints

One of the primary challenges involved limitations imposed by external data sources. The use of free-tier APIs for news ingestion restricted both the number of articles that could be fetched and the historical depth of accessible data. As a result, the volume of company-specific news available for analysis was often limited.

This constraint directly affected downstream modeling, particularly in later weeks where sequence-based models required a minimum amount of continuous data. The experience highlighted the trade-off between ideal model design and realistic data access constraints.

### 5.2 Noise and Ambiguity in News Data

Financial news data is inherently noisy. Keyword-based queries frequently returned irrelevant or weakly related articles, especially for companies with common names. This necessitated additional filtering steps and raised the risk of either over-filtering (leading to data sparsity) or under-filtering (leading to semantic noise).

Balancing relevance and volume proved to be a non-trivial task and reinforced the importance of careful preprocessing in sentiment-driven systems.

## 5.3 Temporal Alignment and Look-Ahead Bias

Ensuring correct temporal alignment between news and market data was a critical challenge. News articles published after market close or during weekends had to be carefully mapped to appropriate trading days to avoid look-ahead bias.

This issue required explicit handling of market hours, weekends, and business day calendars. The challenge emphasized that even minor temporal misalignments can invalidate model results if not handled rigorously.

## 5.4 Model and Environment Constraints

During experimentation, dependency conflicts and environment-related issues were encountered when working with advanced NLP and deep learning libraries. Version compatibilities, hardware limitations, and library requirements restricted the extent to which certain models could be trained or fine-tuned within the given time frame.

As a result, the focus was placed on conceptual understanding and correct pipeline construction rather than extensive hyperparameter tuning or large-scale model training.

## 5.5 Data Sparsity in Sequence Modeling

Sequence-based models such as LSTMs require sufficiently long and continuous time series to learn meaningful temporal patterns. After filtering for relevant news and aligning data correctly, the effective dataset size was sometimes too small to fully exploit the capacity of deep sequence models.

This limitation underscored an important lesson: model complexity must be matched to data availability, and simpler models may be more appropriate under data-scarce conditions.

## 5.6 Scope and Time Constraints

Given the fixed duration of the project and its educational nature, certain aspects such as extensive backtesting, strategy optimization, and robustness evaluation were outside the scope of this work. The project therefore prioritised learning objectives, correctness, and conceptual clarity over production-level performance.

## 5.7 Summary of Challenges

Overall, the challenges encountered were representative of real-world financial data science problems. Rather than detracting from the project, these limitations provided valuable insight into the complexities of building sentiment-driven financial systems and informed more realistic expectations for future work.

# 6 Conclusion

The Market Mood and Moves project provided a comprehensive introduction to the intersection of behavioral finance, natural language processing, and time series modeling. Over the course of four weeks, the project progressed from foundational concepts to the design of an end-to-end sentiment-driven financial analytics pipeline, emphasizing both theoretical rigor and practical constraints.

The early stages of the project highlighted the limitations of classical financial assumptions and motivated the use of sentiment as a proxy for collective market psychology. By studying behavioral finance concepts such as herd behavior, overreaction, and information cascades, the project established a conceptual basis for why news and narrative-driven signals can influence short-term price movements. These insights framed market sentiment not as a replacement for fundamentals, but as a complementary and often leading indicator.

The exploration of modern NLP techniques demonstrated how advances in representation learning have enabled the quantitative analysis of financial text. The transition from static word embeddings to contextual transformer-based models, particularly FinBERT, illustrated the importance of domain adaptation in financial language processing. This phase reinforced the idea that model selection must account for domain-specific semantics and distributional differences rather than relying solely on general-purpose architectures.

Temporal alignment and data integrity emerged as critical considerations throughout the project. The explicit handling of look-ahead bias and trading-day alignment underscored that predictive performance is meaningless if the data pipeline violates real-world causality. These considerations are often underemphasized in academic examples but are central to any system intended for deployment or evaluation in realistic market settings.

The introduction of sequence modeling through LSTM networks extended the analysis from static prediction to dynamic pattern learning. By treating market data as a sequence rather than isolated observations, the project emphasized the role of memory, persistence, and delayed effects in financial time series. The integration of sentiment signals into sequential models further reinforced the multimodal nature of markets, where prices reflect realized outcomes while news sentiment captures evolving expectations.

The final integration phase demonstrated that building a financial analytics system is fundamentally a systems engineering problem. Data sparsity, API limitations, noise in textual data, and the trade-offs between relevance and volume all influenced model behavior. These challenges highlighted that robustness, interpretability, and pipeline correctness are often more valuable than marginal gains in predictive accuracy.

Overall, this project emphasized disciplined reasoning over overfitting, conceptual understanding over blind optimization, and system design over isolated modeling. The learnings from this work extend beyond the specific models used and provide a foundation for approaching real-world financial analytics problems with both technical competence and critical awareness.