Predicting Stock Price Movements from Social Media Sentiment

Jean Bourseau jeanb216@mit.edu
Consti Casper consti99@mit.edu
Lino Valette linoval@mit.edu
Nicholas Wong nicwjh@mit.edu

1 Problem Description

Social media platforms like Twitter and Reddit have become influential sources of retail investor sentiment. The 2021 GameStop short squeeze demonstrated that coordinated retail activity on platforms like r/wallstreetbets can significantly impact stock prices. This raises a natural question: can we predict short-term stock price movements by analyzing social media discussions?

We frame this as a binary classification problem: given all social media posts about a stock on a given trading day, predict whether the stock price will increase or decrease the next trading day. We hypothesize that deep learning models leveraging pre-trained language representations (e.g., BERT) will outperform classical machine learning baselines on TF-IDF features by capturing contextual sentiment patterns that bag-of-words approaches miss.

This task is challenging for several reasons. First, social media text is noisy, containing sarcasm, slang, and informal language that traditional sentiment analysis tools struggle with. Second, the signal-to-noise ratio is inherently low since many factors beyond social sentiment drive stock prices. Third, the temporal nature of financial markets creates regime shifts where patterns learned in one market period may not generalize to another.

The practical implications are significant. Retail investors increasingly rely on social media for investment ideas, and algorithmic trading firms monitor sentiment signals. A successful model could inform trading strategies or risk management. Even negative results would provide valuable insights into the fundamental limits of social media-based prediction.

2 Approach

2.1 Data Collection and Processing

We used the Stock Tweets for Sentiment Analysis and Prediction dataset from Kaggle [1], containing 80,793 tweets about 25 publicly traded companies from September 2021 to September 2022 and combined this with pricing data from Yahoo Finance to create a proprietary dataset. We filtered to 11 stocks with sufficient discussion volume (over 1,000 tweets each): TSLA, TSM, AAPL, AMZN, MSFT, PG, NIO, META, AMD, NFLX, and GOOG.

Our data processing pipeline consisted of four main steps:

1. Text Cleaning: We removed URLs and normalized whitespace while preserving mentions (e.g., @elonmusk) as they may carry predictive signal. Tweets shorter than 20 characters were filtered out.

2. Aggregation: For each (stock, date) pair, we concatenated all tweets from that trading day into a single text sample. This resulted in 2,695 samples across 252 trading days.

3. Label Creation: We pulled daily closing prices from Yahoo Finance using the yfinance library. For each sample, the label is 1 if the stock price increased the next trading day, 0 otherwise. This creates a clear prediction task: today's social sentiment predicts tomorrow's price direction.

4. Train/Val/Test Split: We used a time-based split to avoid lookahead bias. The training set contains the first 70% of trading days (September 2021 to June 2022, 1,909 samples), validation set contains the next 15% (June to August 2022, 395 samples), and test set contains the final 15% (August to September 2022, 391 samples). The test period corresponds to a bearish market regime with only 38% positive labels, while training data is roughly balanced at 48% positive.

2.2 Baseline Model

We established a baseline using Logistic Regression on TF-IDF features. TF-IDF (Term Frequency-Inverse Document Frequency) converts text into numerical vectors where each dimension represents a word or two-word phrase, and values reflect term importance. We used 5,000 features with unigrams and bigrams, fitting the vectorizer only on training data to prevent data leakage.

The naive baseline (always predicting the majority class of the training set) achieves 61.6% accuracy on the test set. Our Logistic Regression model provides a simple but interpretable benchmark that deep learning models should ideally surpass.

2.3 Deep Learning Models

We tested two architectures in order of increasing complexity: an MLP to assess whether nonlinearity helps hand-crafted features, and BERT to assess whether learned representations outperform them.

MLP on TF-IDF Features: We tested whether adding nonlinear layers improves hand-crafted features. We represent each document with 5,000 TF–IDF features (unigrams and bigrams) and feed them into a feedforward network with one to three ReLU hidden layers. We perform a grid search over the number of hidden layers (1–3) and units per layer ({32, 64, 128}), selecting the best configuration by validation accuracy. All models use the Adam optimizer (default learning rate 0.001), binary cross-entropy loss, 20 training epochs, and batch size 32.

BERT with MLP Head: We use pre-trained BERT as a frozen feature extractor, encoding each input text as a 768-dimensional CLS embedding. On top of this, we train a two-layer

MLP head with 128-unit ReLU layers and 0.1 dropout, followed by a sigmoid output for binary classification. Only the MLP head is trained (BERT weights are frozen), using Adam (default learning rate 0.001), binary cross-entropy loss, 10 epochs, and batch size 32.

2.4 Evaluation Metrics

We report three metrics: **Accuracy** (overall correctness), **F1 Score** (precision-recall balance, accounts for imbalance), and **AUC-ROC** (threshold-independent discriminative ability, our primary metric).

3 Results

3.1 Baseline and Model Performance

Table 1: Model Comparison (Test Set)

Model	Accuracy	F1 Score	AUC-ROC
Naive Baseline	61.6%	0.000	0.500
Logistic Regression	50.9%	0.389	0.497
MLP (TF-IDF)	52.2%	0.534	0.538
BERT + MLP Head	48.1%	0.283	0.478

All models performed poorly. The naive baseline achieves 61.6% accuracy by exploiting test set imbalance (38% positive labels during bearish period) but has no discriminative ability (AUC-ROC = 0.500). Logistic Regression achieves near-chance performance (0.497 AUC), indicating minimal signal in TF-IDF features.

The MLP on TF-IDF (52.2% accuracy, 0.538 AUC) shows marginal improvement over Logistic Regression (50.9% accuracy, 0.497 AUC), suggesting that added complexity provides minimal benefit without representation learning.

BERT performed worse than both logistic regression and the MLP (48.1% accuracy, 0.478 AUC), falling below chance. Pre-trained representations from general text corpora fail to transfer to financial social media, and the frozen training approach may prevent effective adaptation. Low F1 scores across all models (0.283-0.534) indicate difficulty identifying positive cases.

3.2 Error Analysis

Three key factors explain poor performance. First, the regime shift between training (48% positive labels, mixed market) and test periods (38% positive, bearish market) prevents generalization across market conditions. Second, aggregating daily tweets discards temporal information: sentiment surges in final trading hours may be more predictive than early posts. Third, the fundamental signal-to-noise ratio is low. Stock prices depend on macroeconomic conditions, earnings, institutional trading, and geopolitical events. Social media represents one weak signal in an extremely noisy system.

4 Lessons Learned

Time-based splitting reveals regime dependence. Our temporal split exposed that models trained in mixed markets fail in bearish periods, a fundamental challenge in financial prediction that random splits would have concealed.

Class imbalance requires robust metrics. The naive baseline's 61.6% accuracy despite 0.500 AUC demonstrates why accuracy alone is misleading. AUC-ROC correctly identified that no model exceeded random guessing meaningfully.

Our hypothesis was rejected: representation learning provides no advantage. Contrary to our expectation, BERT's inferior performance versus TF-IDF features (0.478 vs 0.538 AUC) suggests that general language understanding does not transfer to financial prediction. Domain-specific language and sentiment expressions may require dedicated financial pre-training.

Added complexity provides minimal benefit on hand-crafted features. The MLP's marginal improvement over Logistic Regression (52.2% vs 50.9%) demonstrates that stacking layers on TF-IDF features increases parameters without meaningfully improving generalization.

Signal-to-noise ratio is fundamentally low. Consistent near-chance performance across architectures suggests social media sentiment alone is insufficient for short-term prediction. Our dataset's single-year coverage (September 2021-2022) during unusual conditions further limits generalization.

Key limitations suggest improvements. Future work should: (1) model temporal dynamics explicitly using sequence architectures, (2) fine-tune BERT or use FinBERT rather than frozen features, (3) expand datasets across market regimes and platforms, (4) incorporate multi-task learning to provide richer training signal.

5 Conclusion

This project demonstrates the fundamental difficulty of predicting stock movements from social media text. We hypothesized that pre-trained language models would outperform classical baselines by capturing contextual sentiment, but this hypothesis was rejected. Despite testing multiple approaches, no model achieved meaningful performance, with all barely exceeding random guessing (AUC-ROC near 0.500).

These negative results provide valuable evidence that social media sentiment alone is insufficient for short-term prediction. Low signal-to-noise ratios, regime shifts, and market complexity present significant barriers. However, our dataset's limited temporal coverage, aggregation strategy that discards timing information, and frozen BERT approach suggest caution in generalizing these findings.

Future work should address three key limitations. First, model temporal dynamics explicitly using LSTMs or Transformers with temporal positional encodings to capture sentiment evolution throughout trading days. Second, fine-tune domain-specific models like FinBERT rather than using frozen general-purpose representations. Third, expand datasets across multiple years and market regimes while incorporating additional signals beyond text, such as trading volume, social network structure, and cross-platform sentiment.

The practical implication is clear: investors should not rely solely on social media sentiment for trading decisions. While social media may influence prices, extracting reliable predictive signal requires more sophisticated approaches combining multiple data sources, temporal modeling, and domain-specific adaptation.

Repository Links

We provide implementation in separate notebooks to enable modular execution of the data pipeline, baselines, and deep learning models.

Project Files:

- Google Drive (Complete Project Directory)

Google Colab Notebooks:

- 01_data_processing.ipynb
- 02_mlp_baseline.ipynb
- 03_bert_model.ipynb

GitHub Repository: https://github.com/nicwjh/deep-learning-stock-sentiment

References

[1] Yukhymenko, H. (2022). *Stock Tweets for Sentiment Analysis and Prediction.* Kaggle. Kaggle Link