Université Paris-Est Créteil
Master 2 in Applied Economics, MASERATI / Data Science track

Can Sentiment Analysis Improve the Prediction of Stock Price Direction? An Empirical Study on Apple Inc. (AAPL)

Authors:
Gaël PEFOURQUE, Djibril TRAORE

Supervisor:
Sophie LARUELLE

2024 – 2025

Abstract

This article examines the extent to which integrating sentiment signals extracted from the StockTwits platform can improve daily predictions of Apple Inc. (AAPL) stock price movements. The dataset combines 543 stock market observations (closing price, volume, volatility) and approximately 915,000 StockTwits messages related to AAPL for the period December 31, 2019, to February 27, 2022. Four sentiment analysis methods are used: "Bullish/Bearish" auto-annotations, VADER, FinBERT, and a RoBERTa model finely tuned to StockTwits. Scores are aggregated on a daily basis and weighted by message popularity.

Five modeling scenarios—from a simple "price only" model to "price + sentiment" combinations—are evaluated with three algorithms: SVM, Ensemble SVM (bagging of five SVMs), and LSTM network. The hyperparameters of the SVMs are optimized using sliding walk-forward, while the LSTM is trained on 70% of the data and then tested chronologically on the remaining 30%. Class imbalance is corrected using SMOTE. Performance is measured using the weighted F1 score.

The results show the systematic superiority of LSTM (average F1 = 57.06%) over the SVM ensemble (55.84%) and the simple SVM (54.53%). The best score is achieved with the combination of LSTM + VADER (F1 = 59.91%, +2.7 points compared to the price-only model). An out-of-sample simulation (July 7, 2021 to February 24, 2022) illustrates the economic value of these signals: by investing all capital without transaction costs, the LSTM + FinBERT strategy increases initial capital from $1,000 to $2,115 (+111.5%), more than 100 percentage points better than a simple buy-and-hold approach, which only achieves +11.2%. LSTM + VADER achieves +36.9% over the period. Simple SVMs, which lack sequential memory, remain significantly in deficit.

These results highlight the tangible contribution of sentiment indicators and the relevance of deep learning models for capturing the psychological dynamics of markets. However, the limitations associated with the uniqueness of the asset and the daily horizon suggest that the approach should be extended to other securities, intraday granularities, and multi-asset architectures.

Contents

1 Introduction

On June 5, 2025, TSLA shares suddenly lost some $150 billion in market capitalization after Donald Trump, now openly at odds with Elon Musk, announced that he wanted to "review all federal contracts" awarded to Tesla, SpaceX, and X Corp.

This warning shot came five years after Musk's famous tweet— "Tesla stock price is too high imo"—which had already sent the stock down nearly 10%. Together, these episodes illustrate the power of signals sent on social media: a single message can move billions of dollars in a matter of minutes.

For researchers and investors alike, translating this digital mood into stock market forecasts remains a notoriously difficult problem: the stock market is a highly dynamic, non-linear system, where the psychology of the players carries as much weight as the fundamentals.

Social media—Twitter, StockTwits, and specialized forums—concentrate a wealth of qualitative information that enriches traditional time series data. Several studies have shown that combining these sentiment signals with machine learning models such as SVMs or recurrent neural networks can improve predictions compared to purely technical strategies or those based solely on historical prices.

In this context, our study aims to predict the daily direction of Apple (AAPL) stock by integrating both market data and sentiment indicators from messages posted on the social network Stocktwits. We test several approaches, both in terms of machine learning models and sentiment analysis methods.

The main contributions of our study are as follows:

1. We conduct an experimental comparison of several supervised algorithms (SVM, Ensemble SVM, LSTM) applied to predicting the daily direction of Apple stock.

2. We evaluate the contribution of different sentiment analysis methods (VADER, FinBERT, RoBERTa), which we integrate into the models in the form of daily scores extracted from StockTwits messages.

3. We simulate trading strategies built from the model predictions to compare their performance with that of a passive buy-and-hold strategy over an out-of-sample period.

This thesis is organized as follows: Section 2 provides a literature review on sentiment analysis and stock market prediction. Section 3 describes the data used, distinguishing between market data and messages extracted from StockTwits. Section 4 details the methodology followed, including pre-processing methods, sentiment score calculation, and the predictive models selected. Section 5 presents the results of the different scenarios and simulations of trading strategies. Section 6 discusses the implications of the results obtained and puts them into perspective with previous work. Finally, Section 7 concludes this thesis.

2 Literature review

2.1 Sentiment analysis in finance

Sentiment analysis refers to techniques for automatically extracting opinions, attitudes, and emotions from textual content. In finance, the role of investor sentiment is widely recognized as a key factor influencing financial markets (Baker and Wurgler (2006)). With the rise of online text sources (news, blogs, specialized forums, social networks), it has become possible to measure this market sentiment on a large scale and consider its use to improve stock market forecasting models. Work since the 2000s has suggested a correlation between online opinion climate and stock market activity, justifying the integration of these behavioral factors alongside traditional financial data.

Traditional vs. modern methods: Historically, sentiment analysis in finance has primarily relied on lexicon-based methods. These approaches involve using domain-specific dictionaries of positive/negative words (e.g., the lexicon of Loughran and Mcdonald (2011) for financial reports, or the general-purpose VADER tool designed for social media by Hutto and Gilbert (2014)) to quantify the sentiment of a text. Their advantage lies in their simplicity and interpretability, but they quickly reach their limits when faced with financial jargon or sarcasm. Modern methods, on the other hand, use supervised NLP models and neural networks. For example, a classifier (SVM, logistic regression, etc.) or a deep model (LSTM, BERT) can be trained on labeled texts to classify the sentiment of a document (optimistic vs. pessimistic). These models learn context and often outperform lexicons, especially since the rise of pre-trained transformers. Thus, Batra and Daudpota (2018) built a supervised model that classifies each stock tweet as "bullish" or "bearish" before using these sentiments as inputs to a financial prediction model.
More recently, Ko and Chang (2021) applied BERT, a transformer-based model, to classify the sentiment of financial news articles and stock market messages before feeding them into an LSTM price prediction model. Their results show an average RMSE reduction of 12% compared to using technical data alone, confirming the value of latest-generation NLP models in capturing the nuances of financial language.

Temporal granularity: An important issue in the use of sentiment is temporal alignment with market variations. Investor opinions can change quickly, and their impact on prices can be fleeting. It is therefore crucial to choose the right temporal granularity to aggregate and align sentiment with price data. For example, many studies aggregate sentiment on a daily basis (Batra and Daudpota (2018)), considering each day a dominant polarity (bullish or bearish) based on all messages published that day. This daily aggregation is appropriate if we are looking to predict the closing direction of the following day. Other studies refine the temporal resolution: intraday (hourly) to capture immediate reactions to news, or weekly to smooth out daily noise. Ren et al. (2019) also emphasize that calendar effects (e.g., the day of the week) must be taken into account when constructing reliable sentiment indicators. In short, proper consideration of the temporal dimension (aligning sentiment windows with prediction horizons) is essential to make the most of sentiment analysis in finance.

2.2 Prediction Models Using SVM and LSTM

Several types of predictive models have been used to forecast stock price direction based on financial data and sentiment. Among them, Support Vector Machines (SVMs) and LSTM (Long Short-Term Memory) neural networks are among the most popular. These two families of models have distinct advantages for financial data.

SVMs for stock market prediction: SVMs are supervised classifiers that are well suited to predicting

the direction (up or down) of stock prices, which is treated as a binary classification problem. In finance, an advantage of SVMs is their generalization ability and robustness against overfitting, even on moderately sized samples (Hu et al. (2013)). In fact, training an SVM amounts to solving a convex (quadratic) optimization problem that leads to an optimal global solution, unlike deep neural networks that are subject to non-convex optimizations that can trap them in local minima. In addition, SVMs naturally handle non-linear data through the use of kernel functions, which is useful for capturing complex relationships between financial variables and sentiment.
In fact, several empirical studies have demonstrated the effectiveness of SVMs in predicting market movements. Ren et al. (2019) integrated sentiment variables into an SVM and achieved 89.93% accuracy in predicting the direction of the SSE 50 index, an improvement of 18.6% over the same model without sentiment.
Similarly, Koukaras et al. (2022) tested various classification algorithms (neural networks, decision trees, etc.) to predict the direction of Microsoft stock based on sentiment on Twitter/StockTwits, and SVM proved to be the most effective. In the case of Apple (AAPL), Batra and Daudpota (2018) developed an SVM model combining StockTwits sentiment and stock market indicators, achieving approximately 76.7% accuracy in testing.

These results suggest that SVMs, enriched with sentiment data, are capable of providing useful predictive signals (superior to random or random walk assumptions) on the direction of stock prices. Nevertheless, SVMs do not explicitly model the sequential dimension of the market: they generally operate on aggregated features (e.g., average daily sentiment, price variation) without memory of the past, which motivates the use of dynamic models such as LSTMs.

LSTM for stock market prediction: LSTM networks are a variant of recurrent neural networks designed to model time-series data with long-term dependencies—a crucial asset for financial time series, which exhibit autocorrelations and changing regimes. In practice, LSTM can ingest a time-series of data (e.g., a sequence of daily stock prices and associated sentiment indicators) and learn complex temporal patterns. LSTMs are often used to predict either future prices (regression problem) or the probability of an increase/decrease (classification). Their main advantage is that they can learn the dynamics of a stock over time (trends, cycles, reactions to shocks) better than static models.
Jin et al. (2020) thus propose to predict closing prices by combining an investor sentiment index and an LSTM model with an attention mechanism. They show that the introduction of sentiment from online comments significantly improves prediction accuracy compared to LSTM alone.
Ko and Chang (2021) also confirm the value of incorporating sentiment into sequential models: their hybrid architecture uses BERT to analyze the sentiment of news articles and forum posts, then feeds these signals into an LSTM for price prediction. By comparing several configurations, they find that the use of textual sentiment reduces prediction error (RMSE) and that combining multiple sentiment sources (news + forums) further improves performance, reducing error by an average of 12% compared to an LSTM without sentiment. These examples illustrate that LSTMs enriched with sentiment variables are better at capturing trend reversals and market movements related to investor sentiment. On the other hand, LSTM models require large amounts of data to be trained reliably and can be more difficult to interpret than linear or rule-based models. Despite these challenges, empirical results converge to show that sentiment-informed sequential models outperform their counterparts without this additional information (Jin et al. (2020), Ko and Chang (2021)), fully justifying their use in a financial context.

2.3 Using StockTwits for Stock Market Prediction

Among the available text data sources, the StockTwits platform is a prime choice for gauging investor sentiment in real time. StockTwits is a social network founded in 2009 and dedicated exclusively to discussions about financial markets. It differs from Twitter in its thematic focus: users (traders,

individual investors, etc.) share ideas, opinions, or analyses on stocks and other instruments, all in a short format reminiscent of tweets. Each message can contain cashtags (tickers preceded by the symbol '$') to indicate which asset is being discussed, and, importantly, the author has the option to mark their message as "Bullish" (optimistic) or "Bearish" (pessimistic) about the asset in question. This unique feature provides an explicit indication of the user's sentiment about the stock mentioned. Several empirical studies show that signals extracted from this platform significantly improve the ability to predict stock price direction:

- Batra and Daudpota (2018) train an SVM on 300,000 StockTwits messages related to $AAPL (2010-2017). By aggregating the daily "bullish/bearish" ratio and market variables, they achieve 76.65 % accuracy, outperforming strategies based solely on prices.
- Ren et al. (2019) propose a sentiment index based on Chinese StockTwits posts, weighted by a "day of the week" effect, and then integrate it into an SVM for the SSE 50 index: taking sentiment into account boosts directional accuracy from 71% to 89.9%.
- Sun et al. (2016) propose a stock market prediction model combining text analysis of StockTwits messages and sparse matrix factorization. Using a supervised latent model, they capture the implicit relationships between text messages and S&P 500 stock returns. Their approach reduces prediction error by 22% compared to naive models, demonstrating the potential of specialized, low-noise text data to improve market forecasts.
- Liu et al. (2023) compare various sentiment analyzers (VADER, FinBERT) applied to StockTwits posts on the SPY ETF. With FinBERT and an SVM ensemble, they achieve an F1 score of 0.73 and demonstrate that intraday granularity (12 hours EST) is fine enough to generate actionable trading signals.
- Gupta and Chen (2020) exploit StockTwits' native Bullish/Bearish labels as a free supervised learning dataset to fine-tune a BERT model; when injected into a price prediction LSTM, this sentiment score improves the F1 score by 7 points compared to an LSTM architecture relying solely on market data.

3 Data

3.1 Stock price dataset

We use the Python library yfinance to retrieve, in a single call, the daily adjusted price history (Open, High, Low, Close, Volume) of Apple for the period from December 31, 2019, to February 27, 2022, for a total of 543 observations. This period was specifically chosen to ensure temporal consistency between market data and messages from the StockTwits dataset, covering the period 2020-2022. Table 1 presents an extract of the data retrieved.

Table 1: Extract from the stock price dataset.

Date	Open	High	Low	Close	Volume	Dividends	Stock Splits	Volatility	Return	Log_Return
2020-01-02	71.72	72.78	71.47	72.72	135480400	0.00	0.00	1.31	0.02	0.02
2020-01-03	71.94	72.77	71.78	72.01	146322800	0.00	0.00	0.99	-0.01	-0.01
2020-01-06	71.13	72.62	70.88	72.58	118387200	0.00	0.00	1.75	0.01	0.01
2020-01-07	72.59	72.85	72.02	72.24	108872000	0.00	0.00	0.83	-0.00	-0.00
2020-01-08	71.94	73.71	71.94	73.40	132079200	0.00	0.00	1.76	0.02	0.02

Source: Historical AAPL stock data retrieved via `yfinance` (pypi.org/project/yfinance/). Data covers the period from December 31, 2019 to February 27, 2022.

Figure 1 shows the evolution of Apple's closing price between January 2020 and February 2022. A clear, almost continuous upward trend can be seen: the share price rose from $72.7 to 162.2$, an increase of more than 120%. This dominance of "bullish" days introduces a class imbalance in our target variable (day + versus day -), which justifies:

1. The use of robust metrics, such as the weighted F1 score, to evaluate the quality of predictions beyond simple accuracy.

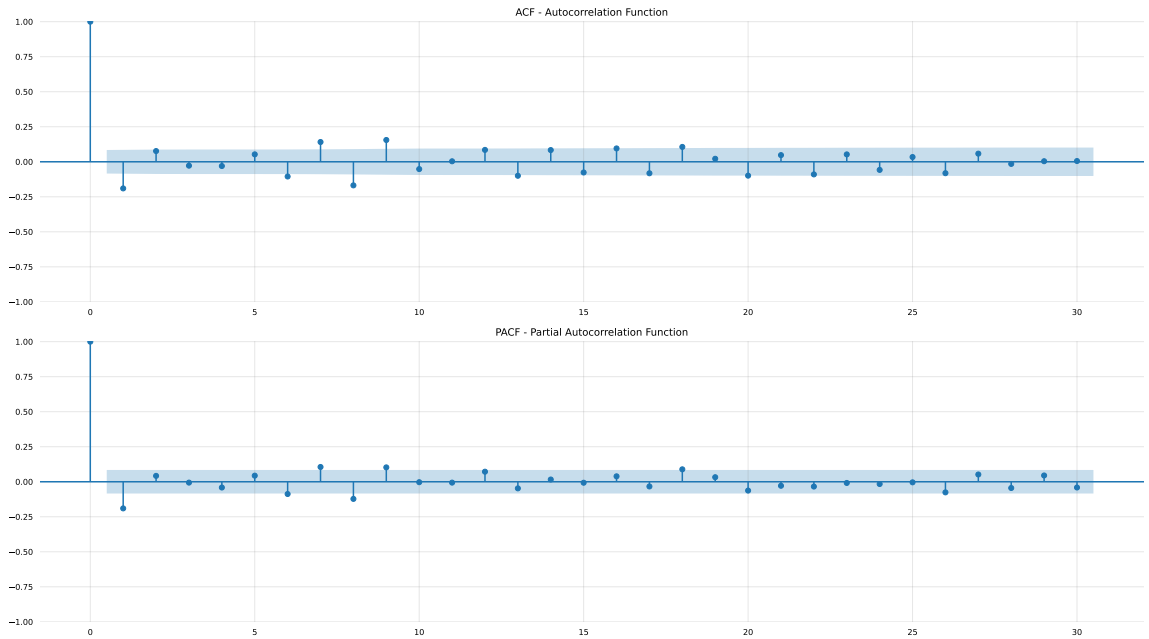
2. The use of oversampling techniques to counterbalance the overrepresentation of positive days and prevent a naive model from simply predicting "rise" systematically.

Furthermore, the analysis of autocorrelation functions (ACF) and partial autocorrelation functions (PACF) presented in Figure 2 highlights a very rapid decrease in coefficients beyond lag 1. This structure indicates weak linear memory: the historical price series does not have sufficient dependence to explain, on its own, the sign of daily variations. This justifies the addition of exogenous variables to overcome this limitation and improve the reliability of our forecasts.

Figure 1: Apple Inc. (AAPL) stock closing price between December 31, 2019, and February 27, 2022.

Source: Historical AAPL stock data retrieved via `yfinance` (pypi.org/project/yfinance/). Data covers the period from December 31, 2019 to February 27, 2022.

Figure 2: ACF and PACF of AAPL log returns between December 31, 2019, and February 27, 2022.



Source: Historical AAPL stock data retrieved via `yfinance` (pypi.org/project/yfinance/). Data covers the period from December 31, 2019 to February 27, 2022.

3.2 StockTwits dataset

We retrieved a dataset of 915,000 messages posted on StockTwits related to $AAPL from Kaggle, an online platform specializing in data science that allows users to participate in machine learning competitions and share datasets.
Table 2 provides a quantitative and qualitative overview of our corpus of StockTwits messages, which will serve as a source of sentiment indicators for predicting the direction of AAPL's share price. The period observed runs from December 31, 2019, to February 27, 2022, representing 790 trading days. Nearly 915,000 messages were collected, with an average of 1,158 messages per day (standard deviation = 1,047), reflecting a high variability in activity from one session to another. The busiest day (October 13, 2020) saw more than 8,500 posts, while the quietest (February 27, 2022) saw only 91, no doubt due to the collection of messages being stopped before the end of the day. The high heterogeneity of message volume means that the quantity of messages must be taken into account, in addition to the sentiment analysis itself, when adding them to models.

Furthermore, 35.3% of messages are labeled "Bullish" and 11.2% "Bearish," while 50.0% of posts do not have an explicit label provided by users at the time of publication. The significant proportion of unlabeled messages requires the use of sentiment analysis methods (VADER, FinBERT, RoBERTa) capable of classifying these texts in order to produce a continuous and comprehensive daily score.

Finally, the average length is 106.5 characters per message. This short format, often rich in abbreviations and emojis, justifies the use of pre-trained NLP tools specialized in the financial field (e.g., FinBERT) or robust to informal texts (VADER), in order to best capture the tone and emotional nuance of investors.

Table 2: Descriptive Statistics of the StockTwits data.

	Observed value
Covered period	2019-12-31 → 2022-02-27
Total number of messages	914962
Number of days covered	790
Average messages per day	1158.18
Median messages per day	929.00
Standard deviation of messages/day	1046.98
Most active day	2020-10-13 (8506 msg)
Least active day	2022-02-27 (91 msg)
% of "Bullish" messages	35.29
% of "Bearish" messages	11.24
% of messages without label	50.04
Average message length (characters)	106.52

Source: Stocktwits messages dataset hosted on Kaggle (kaggle.com). Data covers the period from December 31, 2019 to February 27, 2022.

Figure 3 shows three word clouds from messages posted on StockTwits. Across the entire corpus, we find generic market-related terms ("market," "today," "stock," "apple," "amzn"), reflecting investors' constant interest in major stocks and the macroeconomic context.

In contrast, the "Bullish" and "Bearish" subcorpora are distinguished by the increased presence of strategic terms ("buy," "calls," 'puts') and symbols of other securities ("spy," "tsla," "qqq")—signs

that investors situate their opinion of Apple in a broader universe of derivatives and indices.

The lexical diversity—stocks, options, indices, temporality ("today," "week," "tomorrow")—attests to the fact that investor sentiment incorporates both a short-term view (scalping, day trading) and a medium-term view (anticipation of quarterly results). The distinction between bullish and bearish keywords validates the use of sentiment analysis algorithms to transform these texts into coherent quantitative scores.

Figure 3: Comparison of word clouds generated from StockTwits messages based on self-annotated sentiment.

Source: Stocktwits messages dataset hosted on Kaggle (kaggle.com). Data covers the period from December 31, 2019 to February 27, 2022.

Table 3 presents a summary of the sentiment annotations for the different methods explored in our study. The Base, FinBERT, and roBERTa methods show a bullish bias (+70% bullish), which can lead predictive models to systematically favor upward movement. VADER introduces a significant proportion of neutral messages (44%), but can dilute the directional signal.

Table 3: Summary of the sentiment analysis results.

	Bullish	Bearish	Neutral	Bullish (%)	Bearish (%)	Neutral (%)
Base	345900	107549	0	76.3	23.7	0.0
Vader	317381	193122	404459	34.7	21.1	44.2
Finbert	710612	78121	126229	77.7	8.5	13.8
Roberta	650051	264911	0	71.0	29.0	0.0

Source: Stocktwits messages dataset hosted on Kaggle (kaggle.com). Data covers the period from December 31, 2019 to February 27, 2022.

The Figure below (Figure 4) superimposes, over the period January 2020 – March 2022, the smoothed series of the four daily sentiment scores applied to StockTwits messages.

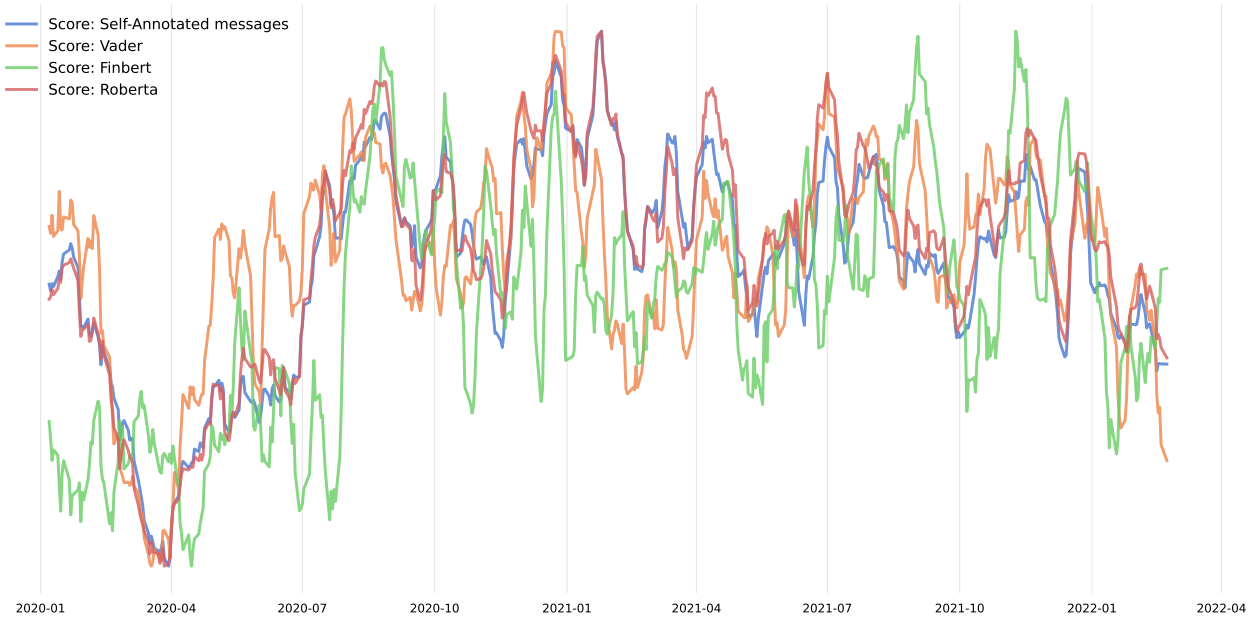
1. FinBERT (green) stands out with wider and occasionally more erratic variations: there are more pronounced upward and downward spikes, indicating increased sensitivity to certain financial keywords.

2. VADER (orange) and RoBERTa (red) are smoother and sometimes slightly out of sync: VADER is sensitive to linguistic context (emoticons, intensifiers), while RoBERTa benefits from specific training on StockTwits.

3. La Base (blue) — a raw aggregate of self-annotations — acts as a reference signal, less noisy than FinBERT but more so than VADER/RoBERTa.

The four approaches extract a consistent directional signal, aligned with the major market phases, but they differ in amplitude, stability, and timing. This justifies empirically testing each approach in prediction models as well as in out-of-sample simulation of trading strategies.

Figure 4: Smoothed daily sentiment scores for AAPL-related Stocktwits messages, based on different sentiment models (self-annotated messages, VADER, FinBERT, and RoBERTa)



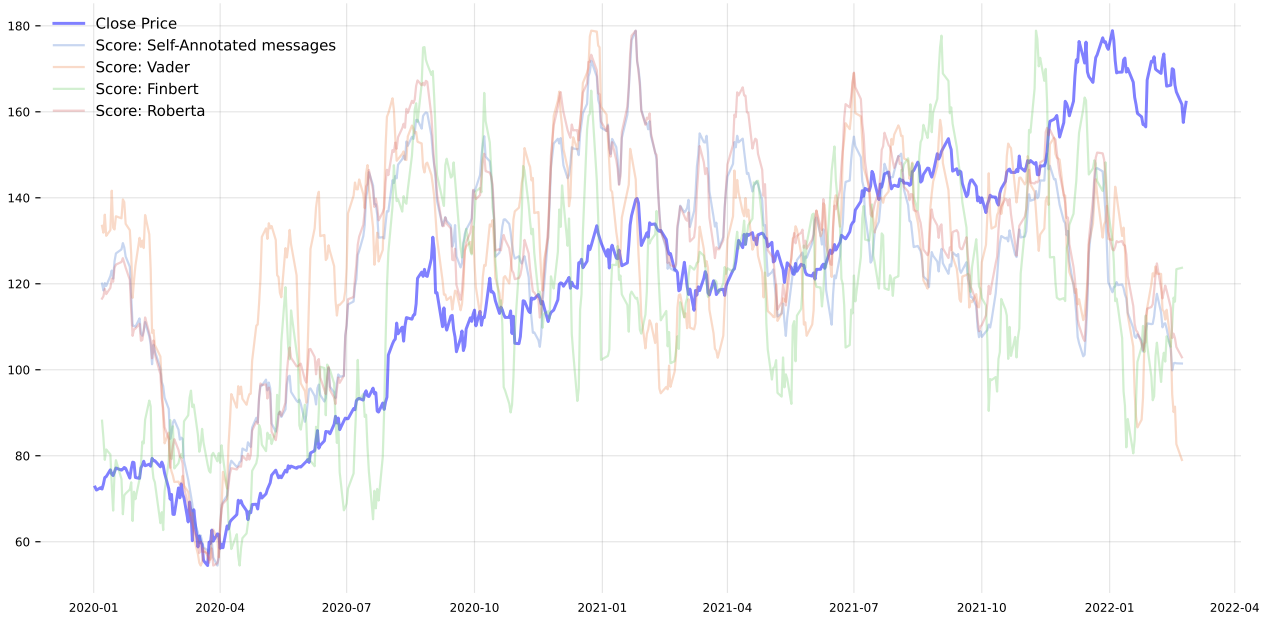
Note: Sentiment scores are shown as 7-day centered rolling averages with independent Y-axes.

Sources: Historical AAPL stock data retrieved via `yfinance` (pypi.org/project/yfinance/), and Stocktwits messages dataset hosted on Kaggle (kaggle.com). Data covers the period from December 31, 2019 to February 27, 2022.

In this merged Figure (Figure 5), we observe that the four sentiment curves (Base, VADER, FinBERT, RoBERTa), although noisier, broadly follow the major movements in the closing price: the trough in spring 2020, the peak in early 2021, and the decline in early 2022. Furthermore, we note that sentiment signals—particularly VADER and RoBERTa, but also some FinBERT spikes—often precede price changes by a few days. This ability to "anticipate" reversals validates the idea of incorporating

these scores as lagged exogenous variables in our SVM or LSTM models to improve daily direction prediction where price dynamics alone show limited autocorrelations.

Figure 5: AAPL closing price and smoothed sentiment scores from Stocktwits messages between January 2020 and March 2022, based on different models (self-annotated messages, VADER, FinBERT, and RoBERTa).



Note: Sentiment scores are shown as 7-day centered rolling averages with independent Y-axes.

Sources: Historical AAPL stock data retrieved via `yfinance` (pypi.org/project/yfinance/), and Stocktwits messages dataset hosted on Kaggle (kaggle.com). Data covers the period from December 31, 2019 to February 27, 2022.

4 Methodology

4.1 Support Vector Machine

The Support Vector Machine (SVM), developed by Cortes and Vapnik (1995), is a binary classifier that aims to find the hyperplane that maximizes the margin between two classes.

We are looking for a hyperplane

$$w^\top x + b = 0 \tag{1}$$

such that, for all support vectors,

$$y^{(i)}\left(w^\top x^{(i)} + b\right) \geq 1, \tag{2}$$

and that the margin

$$\frac{2}{\|w\|} \tag{3}$$

is maximized, where w is the vector of weights for each variable and b is the intercept (bias).

To tolerate classification errors, we introduce relaxation variables $\xi_i \geq 0$ and a parameter $C > 0$:

$$\begin{aligned} \min_{w,b,\xi} \quad & \tfrac{1}{2}\|w\|^2 \; + \; C \sum_{i=1}^{n} \xi_i, \\ \text{under constraint} \quad & y^{(i)}\big(w^\top x^{(i)} + b\big) \geq 1 - \xi_i. \end{aligned} \tag{4}$$

The value of parameter C controls the error tolerance: the larger C is, the less the model tolerates errors, bringing it closer to a strict margin SVM when C tends to infinity. Conversely, a smaller C allows for more errors, thus increasing the flexibility of the model.

The dual problem is written as

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^{n} \alpha_i \; - \; \tfrac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j \, y_i y_j \, K(x_i, x_j), \\ \text{under constraint} \quad & \sum_{i=1}^{n} \alpha_i \, y_i = 0, \quad 0 \leq \alpha_i \leq C, \end{aligned} \tag{5}$$

where $K(x_i, x_j)$ is the kernel (linear, polynomial, RBF, etc.).

The prediction for a new example x is

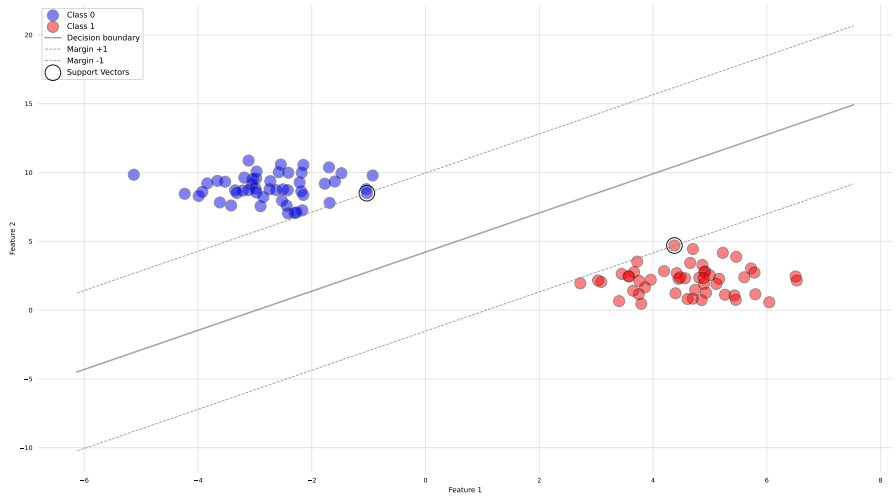
$$\hat{y} \; = \; \operatorname{sign}\Big(\sum_{i=1}^{n} \alpha_i^* \, y_i \, K(x_i, x) \; + \; b^* \Big). \tag{6}$$

Choice of kernel and hyperparameters:

- Linear kernel: $K(x, x') = x^\top x'$.
- RBF kernel: $K(x, x') = \exp\big(-\gamma \|x - x'\|^2\big)$.
- Hyperparameters: C and γ chosen by cross-validation.

SVM, with its margin maximization and flexibility via kernels, is particularly well suited to the combined analysis of financial variables and sentiment variables: it offers good generalization in high dimensions and allows class imbalances to be managed effectively.

Figure 6: Illustration of the optimal separating hyperplane on synthetic data, maximizing the margin between two linearly separable classes.



4.2 SVM ensemble with bagging

To reduce variance and improve the robustness of an SVM, a set can be constructed using bagging (Bootstrap Aggregating). The idea is to generate M training datasets using bootstrap sampling, train M independent SVMs on them, and then combine their decisions by majority vote.

Principle

1. At each iteration $m = 1, \ldots, M$:
   - A subsample of size n is randomly selected, with replacement, from the training set.
   - An SVM is trained on this subsample:

$$f_m(x) = \text{sign}\Big(\sum_{i \in \mathcal{I}_m} \alpha_i^{(m)} y_i \, K(x_i, x) + b^{(m)}\Big).$$

2. For a new example x, the predictions are aggregated by majority vote:

$$\hat{y} = \text{maj}\{\, f_m(x)\}_{m=1}^{M}.$$

Advantages of bagging are:

- **Variance reduction**: each classifier sees a different version of the data.
- **Increased robustness** to outliers and overfitting.
- **Ease of implementation** by simply wrapping the SVM in a `BaggingClassifier`.

4.3 Long Short-Term Memory

4.3.1 Theoretical Background on LSTM

Long Short-Term Memory (LSTM) are recurrent neural network architectures designed to capture long-term dependencies in time series. At each time step t, we have an input vector x_t, a previous hidden state h_{t-1}, and a memory cell c_{t-1}. The internal calculations of an LSTM cell are as follows:

$$f_t = \sigma\big(W_f[x_t, h_{t-1}] + b_f\big) \quad \text{(forget gate)} \tag{7}$$
$$i_t = \sigma\big(W_i[x_t, h_{t-1}] + b_i\big) \quad \text{(input gate)} \tag{8}$$
$$\tilde{c}_t = \tanh\big(W_c[x_t, h_{t-1}] + b_c\big) \quad \text{(candidate cell state)} \tag{9}$$
$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \quad \text{(cell update)} \tag{10}$$
$$o_t = \sigma\big(W_o[x_t, h_{t-1}] + b_o\big) \quad \text{(output gate)} \tag{11}$$
$$h_t = o_t \odot \tanh(c_t) \quad \text{(hidden state)} \tag{12}$$

where σ is the sigmoid function, $\odot$ is the term-by-term product, and W_*, b_* are the learned weights and biases.

4.3.2 Specification of the PyTorch Architecture

In our implementation, we use the `torch.nn.LSTM` class, which encapsulates the above equations. The `LSTMClassifier` model is defined by:

```
class LSTMClassifier(nn.Module):
    def __init__(self,
        input_size: int,
        hidden_size: int,
        num_layers: int = 1,
        dropout: float = 0.0):
    super().__init__()
    self.lstm = nn.LSTM(
        input_size=input_size,
        hidden_size=hidden_size,
        num_layers=num_layers,
        batch_first=True,
        dropout=dropout
)
self.fc = nn.Linear(hidden_size, 2)
```

where:

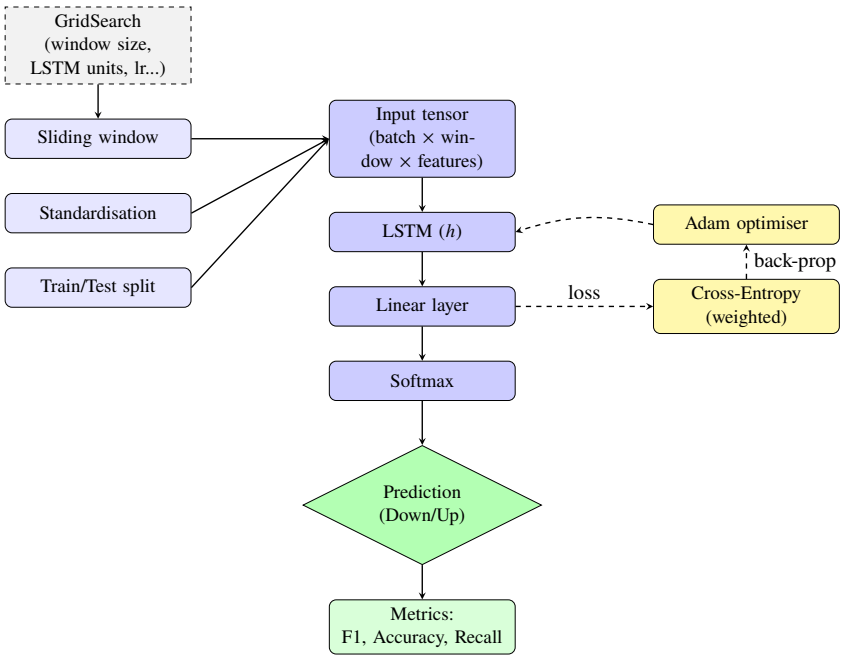
- `input_size` is the dimension of vector x_t,
- `hidden_size` is the dimension of the hidden states h_t and cells c_t,
- `num_layers` determines the number of stacked LSTM layers,
- `dropout` is the probability of dropout between layers, applicable if `num_layers` > 1.

The forward is written as follows:

```
def forward(self, x):
    # x shape: (batch, seq_len, input_size)
    _, (hn, cn) = self.lstm(x)
    # hn shape: (num_layers, batch, hidden_size)
    last_hidden = hn[-1] # (batch, hidden_size)
    logits = self.fc(last_hidden) # (batch, 2)
    return logits
```

We retrieve $h_n = hn[-1]$, the hidden state of the last layer at the last time step, which we transform into two *logits* via a linear output layer. Due to the limited size of our dataset, we opted for a single-layer LSTM architecture, capable of effectively modeling temporal dependencies while limiting the risk of overfitting.

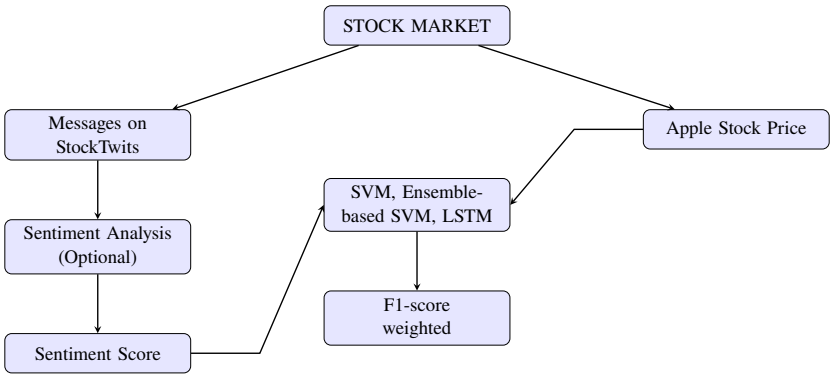
Figure 7: Architecture of the LSTM pipeline using a rolling window approach and supervised learning.



Sources: Historical AAPL stock data retrieved via `yfinance` (pypi.org/project/yfinance/), and Stocktwits messages dataset hosted on Kaggle (kaggle.com). Data covers the period from December 31, 2019 to February 27, 2022.

4.4 Building the Prediction Pipeline

Figure 8: Pipeline architecture for predicting stock market movements using sentiment and price data.



Sources: Historical AAPL stock data retrieved via `yfinance` (pypi.org/project/yfinance/), and Stocktwits messages dataset hosted on Kaggle (kaggle.com). Data covers the period from December 31, 2019 to February 27, 2022.

4.4.1 Data collection

Figure 8 shows the architecture used to predict the direction of change in Apple's closing share price: We assemble financial and textual data from the two databases we collected, covering the period from December 31, 2019, to February 27, 2022.

4.4.2 Preprocessing for Sentiment Analysis

Prior to sentiment analysis, we preprocess the messages:

1. Removal of URL links.
2. Formatting of incorrectly encoded characters.
3. Replacement of hashtags "#" and cahstags "$" with "hashtag_" and "cashtag_" respectively, to improve text processing.
4. Replacement of "@" mentions with "mention_" followed by the name of the user mentioned.
5. Replacement of emojis with their textual description using the `emoji` library in Python, in order to make the text more compliant with NLP model requirements while retaining the information provided by the emojis.

4.4.3 Sentiment score computation

The four sentiment classification methods we use are:

1. **Base**: These are messages auto-annotated by users when they post their messages. 50% of messages are therefore unannotated, which introduces a bias in the analysis of annotated messages alone. For this reason, we will compare this approach with the following models:

2. **VADER**: VADER is a sentiment analysis tool that uses a combination of a sentiment lexicon and rules to evaluate text, specifically designed to accurately capture the informal language and expressions commonly found in social media. (Hutto and Gilbert (2014))

3. **FinBERT**: FinBERT is trained on a collection of financial texts, including news articles, analyst reports, earnings call transcripts, and regulatory filings such as 10-Ks. These texts are labeled with financial sentiment—positive, negative, or neutral—so the model learns how sentiment is typically expressed in financial contexts. It is built on top of the original BERT model and fine-tuned to understand the specific language and tone used in finance. Araci (2019)

4. **RoBERTa**: The model we use is a version of RoBERTa trained specifically on approximately 3.2 million StockTwits messages annotated as "Bullish" or "Bearish."

We use the method proposed by Antweiler and Frank (2004) to calculate the daily sentiment score of the two-class sentiment analysis models (Base, RoBERTa):

$$S_t = \ln\left(\frac{1 + M_t^{\text{pos}}}{1 + M_t^{\text{neg}}}\right) \tag{13}$$

Where S_t is the sentiment score, M_t^{pos} is the number of messages classified as "Bullish," and M_t^{neg} is the number of messages classified as "Bearish."

For three-class scores (VADER, FinBERT), we use the method proposed by Hiew et al. (2019):

$$S_t = \ln\left(\frac{M_t^{\text{pos}} - M_t^{\text{neg}}}{M_t^{\text{pos}} + M_t^{\text{neu}} + M_t^{\text{neg}}}\right) \tag{14}$$

With S_t being the sentiment score and M_t^{pos}, M_t^{neu}, M_t^{neg} respectively the number of messages classified as "Bullish", 'Neutral' and "Bearish".

Like ponderation: To take into account the influence of a message shared on Stocktwits and therefore its impact on the financial markets, we choose to weight messages according to the number of likes received:

$$W = \sqrt{1 + N_{\text{likes}}} \tag{15}$$

Where N_{likes} is the number of likes received per message and W is the weight applied to the message in the daily sentiment score. This method allows us to better take into account the impact of posted messages, while controlling the influence of messages with many likes by applying a square root.

4.4.4 Model building

We construct the target variable y as a binary variable indicating the direction of the next day's closing price:

$$y = \begin{cases} 1, & \text{Close Price } t+1 \geq \text{Close Price } t \\ 0, & \text{Close Price } t+1 < \text{Close Price } t \end{cases}$$

This transforms a regression problem into a classification problem, as we focus on predicting the direction of change in the closing price rather than the actual price.

- `Close`: The closing price on day t
- `nb_tweets`: The number of messages posted on day t

- `nb_tweets_diff`: The number of messages posted on day $t-1$
- `score_scenario`: The sentiment score calculated for day t
- `score_scenario_diff`: The sentiment score calculated for day $t-1$

We run five scenarios to compare the results of each approach:

1. **Price-only model (reference)**:

   Only historical closing price data (`Close`) is used as explanatory variables to predict future changes in the share price. No sentiment indicators are taken into account. This model is our baseline reference.

2. **Addition of user-annotated sentiments**: In addition to historical prices, we incorporate sentiment indices from messages annotated by Stocktwits users themselves. More specifically, we use the daily average sentiment score and the number of messages at times t and $t-1$, i.e., four new variables.

3. **Classification of messages with VADER**: In order to increase the amount of sentiment data used, all messages, including unannotated ones, are classified according to their tone using the VADER tool. Sentiment indices are then calculated based on the proportions of messages deemed positive, neutral, or negative at times t and $t-1$.

4. **Classification of messages with FinBERT**: All Stocktwits messages are automatically labeled by the FinBERT model, which specializes in sentiment analysis in finance. As before, sentiment indices are constructed from the proportions of messages deemed positive, neutral, or negative at times t and $t-1$.

5. **Classification of messages with RoBERTa fine-tuned on Stocktwits**: This last scenario is based on the `roberta-base-Stocktwits-finetuned` model, trained specifically on Stocktwits messages. Sentiment scores are calculated from the proportions of messages classified as bullish or bearish at times t and $t-1$.

4.4.5 Evaluation metrics

The performance of the models is assessed using two global indicators—accuracy and weighted F1-score—as well as precision, recall, and F1-score measures calculated for each of the two classes (bullish = 1, bearish = 0).

Overall accuracy Accuracy (Acc) measures the proportion of correctly classified observations:

$$\text{Acc} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{16}$$

where, for each class c,

- TP (True Positives) is the number of examples in the positive class that are correctly predicted;
- TN (True Negatives) is the number of examples in the negative class that are correctly predicted;
- FP (False Positives) is the number of negative examples predicted as positive;
- FN (False Negatives) is the number of positive examples predicted as negative.

Precision, recall, and F1-score per class For each class $c \in \{0, 1\}$, we define:

$$\text{Precision}_c = \frac{\text{TP}_c}{\text{TP}_c + \text{FP}_c}, \tag{17}$$

$$\text{Recall}_c = \frac{\text{TP}_c}{\text{TP}_c + \text{FN}_c}, \tag{18}$$

$$\text{F1}_c = 2 \times \frac{\text{Precision}_c \times \text{Recall}_c}{\text{Precision}_c + \text{Recall}_c}. \tag{19}$$

Here, TP_c, FP_c, FN_c denote the number of true positives, false positives, and false negatives for class c, respectively.

Weighted F1-score The weighted F1-score (F1_w) aggregates the F1-scores of each class, taking into account their proportion in the dataset:

$$\text{F1}_\text{w} = \sum_{c \in \{0,1\}} \underbrace{\frac{N_c}{N}}_{w_c} \times \text{F1}_c \tag{20}$$

where N_c is the number of samples in class c in the test set and $N = N_0 + N_1$ is the total number of samples. The weight $w_c = N_c/N$ thus reflects the relative importance of each class.

These metrics provide a comprehensive assessment of the model's ability to distinguish between days of increase (1) and days of decrease (0), while compensating for any class imbalances using the weighted F1 score.

4.5 Model Implementation

We implemented two sequential classification approaches—SVM and LSTM—using a two-phase methodology for each of the five scenarios tested. First, the models were trained and tested over the period from December 31, 2019, to July 6, 2021. Second, the calibrated models are tested out-of-sample from July 7, 2021, to February 24, 2022, through a simulation of long/short trading strategies.

4.5.1 Hyperparameter Optimization

For SVM and Ensemble SVM models, the tuning phase is based on a walk forward procedure: at each date t, the model is trained on the previous w days, then tested on the prediction for day $t + 1$. All hyperparameters are evaluated using the weighted F1 score across all windows considered, and the optimal configuration is the one that maximizes this statistic.

For the LSTM model, due to constraints related to the training time of a deep learning model, the LSTM is not retrained every day, but rather separates the dataset into train/test chronologically: 70% train and 30% test. Thus, even if the model does not follow a walk forward procedure, it is never trained with future data.

Support Vector Machine The exploration grid covers window lengths $w \in \{80, 81, \ldots, 119\}$ days and the regularization parameter $C \in \{0{,}1, 1, 10\}$. The RBF kernel adopts automatic adjustment of the parameter γ ("scale"). To correct class imbalance, the training set is preprocessed by SMOTE, then

standardized using "RobustScaler." Performance is measured in terms of accuracy, weighted F1-score, and precision, recall, and F1-score for classes 0 and 1. The final selection retains the configuration offering the best weighted F1-score.

SVM ensemble In addition to the simple model, bagging of $M = 5$ SVM estimators is applied to form an SVM ensemble model.

Long Short-Term Memory The search for the LSTM covers windows of $w \in \{230, 231, \ldots, 249\}$ days and varies three hyperparameters: the number of hidden units ($\{15, 25\}$), the dropout rate ($\{0, 0.2\}$), and the learning rate ($\{10^{-3}, 10^{-4}\}$). Each model is trained with weighted CrossEntropy loss and the Adam optimizer, over 50 epochs and a batch size of 16. All features are normalized before training. The final selection retains the configuration offering the best weighted F1-score.

4.5.2 Out-of-Sample Evaluation and Trading Simulation

The out-of-sample evaluation period runs from July 7, 2021, to February 24, 2022 (161 trading days). At each date t, the models with the best hyperparameters over the previous period (December 31, 2019, to July 6, 2021) provide a prediction

$$s_t = \begin{cases} 1, & \text{if an increase is expected,} \\ 0, & \text{if a decrease is expected.} \end{cases}$$

On this basis, we compare two active trading strategies and a "buy and hold" benchmark strategy:

1. **Frictionless "full" strategy**: Daily investment of 100% of capital ($\alpha = 1$), with no transaction fees ($\tau = 0$).
2. **Realistic "full" strategy**: Daily investment of 50% of capital ($\alpha = 0.5$), transaction fees of $\tau = 0{,}0005$ (0.05 %) applied at opening and closing.
3. **Buy and hold**: Long position held every day without signal adjustment, no active reinvestment and no fees.

For both active strategies, the gross return is defined by

$$r_t = \begin{cases} \dfrac{P_{t+1} - P_t}{P_t}, & \text{if } s_t = 1 \text{ (long)}, \\ \dfrac{P_t - P_{t+1}}{P_t}, & \text{if } s_t = 0 \text{ (short)}, \end{cases}$$

then the net PnL is written as

$$\mathrm{PnL}_t = \alpha\, C_t\, r_t \;-\; 2\,\tau\,\alpha\, C_t,$$

where C_t is the capital available at the close of t. The capital is then updated by

$$C_{t+1} = C_t + \mathrm{PnL}_t.$$

Each trading day results in the recording of the following variables:

$$\big\{\mathrm{Date}_t,\ \mathrm{Close}_t,\ \mathrm{Close}_{t+1},\ s_t,\ r_t,\ \text{opening fees},\ \text{closing fees},\ C_t,\ C_{t+1}\big\}.$$

5 Results & Trading simulation

5.1 Results

We executed each LSTM configuration on 10 distinct random seeds to account for the variance introduced by weight initialization and shuffling. This procedure aims to obtain a reliable and robust estimate of the model's performance, avoiding an exceptional result (too optimistic or too pessimistic) due to a particularly "lucky" or "unlucky" seed.

Thus, for each scenario, Table 4 shows both the average F1 score (over these 10 runs) and its standard deviation, the latter measuring the dispersion of the scores around the average. A low standard deviation indicates high model stability regardless of the initial draw, while a higher standard deviation indicates increased sensitivity to starting conditions. For example, scenario 3 (integration of VADER) not only displays the best average F1 score (0.5991), but also has the lowest variability (standard deviation 0.0206), ensuring both superior and consistent performance across all test seeds.

Table 4: Mean and standard deviation of LSTM F1-scores across 10 random seeds for each scenario.

Scenario	Mean F1-Score	Standard Deviation
1	0.5723	0.0264
2	0.5694	0.0242
3	0.5991	0.0206
4	0.5734	0.0347
5	0.5390	0.0548

Sources: Historical AAPL stock data retrieved via `yfinance` (pypi.org/project/yfinance/), and Stocktwits messages dataset hosted on Kaggle (kaggle.com). Data covers the period from December 31, 2019 to February 27, 2022.

Table 5: F1-score comparison of SVM, ensemble SVM, and LSTM models across the five scenarios.

Scenario	SVM	Ensemble SVM	LSTM	Mean per scenario
1	51.91%	53.49%	57.23%	54.21%
2	55.03%	56.4%	56.94%	56.12%
3	55.51%	56.69%	59.91%	57.37%
4	55.15%	55.93%	57.34%	56.14%
5	55.06%	56.71%	53.9%	55.22%
Mean per model	54.53%	55.84%	57.06%	55.81%

Sources: Historical AAPL stock data retrieved via `yfinance` (pypi.org/project/yfinance/), and Stocktwits messages dataset hosted on Kaggle (kaggle.com). Data covers the period from December 31, 2019 to February 27, 2022.

Table 5 compares the F1 scores obtained by the three proposed models—classical SVM, Ensemble SVM, and LSTM—on the five enrichment scenarios described in Section 4. Each score corresponds

to the average F1 score (%) calculated on the "increase" and "decrease" classes for Apple over the period from December 31, 2019, to July 6, 2021. The last column gives the average of the three models per scenario, and the last row gives the average of each model across all scenarios.

First, the LSTM architecture stands out overall as the best performer, with an average F1 score of 57.06%, compared to 55.84% for the Ensemble SVM and 54.53% for the simple SVM. This superiority highlights the ability of recurrent networks to capture temporal dynamics and sequential dependencies in both price series and text signals.

Scenario 1 (Price only) The LSTM achieved 57.23%, outperforming the Ensemble SVM (53.49%) by +3.74 points and the simple SVM (51.91%) by +5.32 points. This initial difference highlights the methodological advantage of LSTM for modeling the intricate variations of financial time series without text mining.

Scenario 2 (Self-annotated sentiments) The addition of self-reported "Bullish/Bearish" labels improves all three approaches, with a gain of +3.12 points for the simple SVM and +2.91 points for the ensemble model. LSTM also improves to 56.94%, but the gap with the ensemble model narrows to 0.54 points, demonstrating a convergence in performance when the textual signal is rudimentary.

Scenario 3 (VADER scores) The integration of VADER sentiment scores propelled LSTM to its peak of 59.91%, a gain of +2.97 points compared to scenario 2 and +2.58 points over the best SVM (56.69%). This confirms the value of a fine-grained lexical analysis that can be adapted to the informal style of financial messages.

Scenario 4 (FinBERT embeddings) The use of FinBERT brings a moderate benefit: LSTM achieves 57.34%, a slight increase compared to scenario 1 (+0.11 points), while SVMs stagnate around 55–56%. This suggests that, without specific adjustments, the semantic richness of financial embeddings is not fully exploited by linear classifiers.

Scenario 5 (RoBERTa embeddings) Conversely, RoBERTa fine-tuned on StockTwits penalizes LSTM (53.90%), while Ensemble SVM takes advantage of this type of embedding to reach 56.71%.

The results in Table 5 show that the combination of **LSTM + VADER** offers the best compromise between performance and robustness for predicting stock price direction. SVMs, in particular Ensemble SVMs, nevertheless remain competitive alternatives when the textual signal is limited or poorly suited, and have the advantage of being much less complex to train than LSTMs.

5.2 Trading simulation based on model predictions

5.2.1 Support Vector Machine

Figure 9 illustrates, for each of the five SVM strategies corresponding to scenarios 1 to 5, the evolution of an initial capital of $1,000 over the period July 7, 2021 – February 24, 2022, according to the two trading strategies:

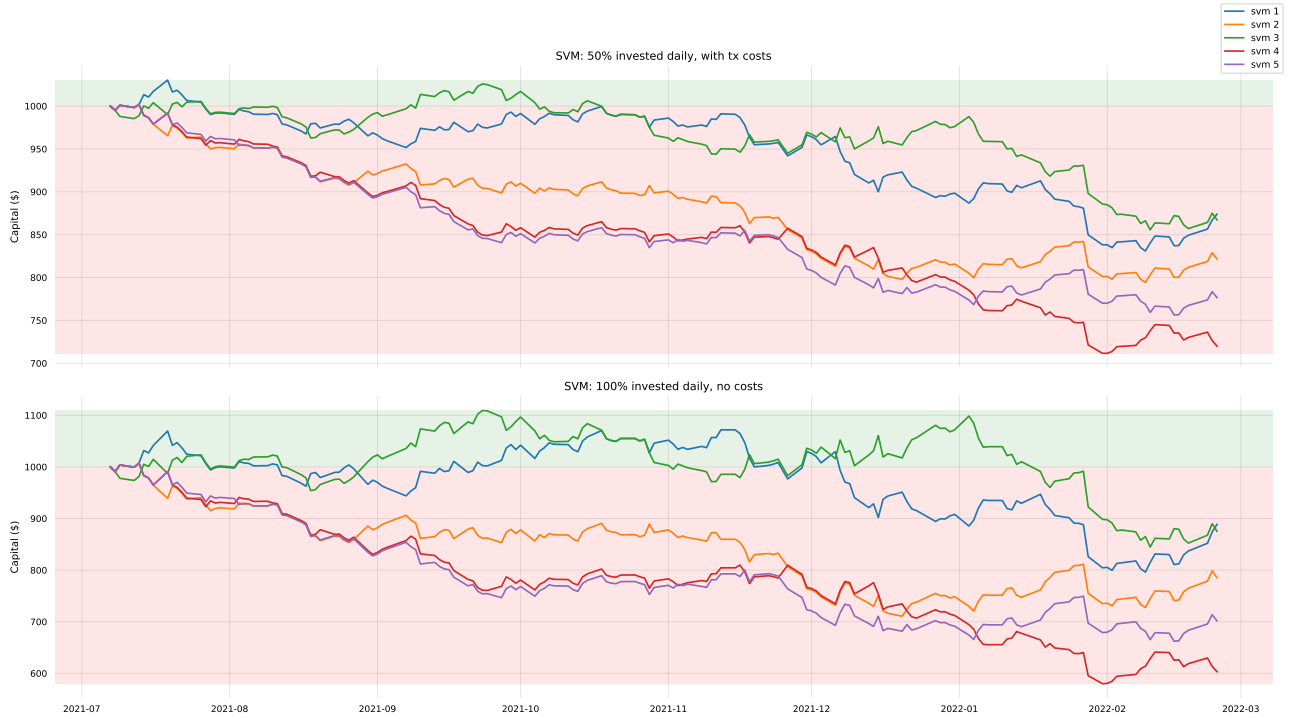
- 50% daily exposure with transaction costs (upper panel).
- 100% daily exposure without costs (lower panel).

In each panel, the green area corresponds to capital ≥ \$1,000 (net gain), and the red area corresponds to capital ≤ \$1,000 (net loss).

Upper tranche: 50% invested with costs Under this configuration, all portfolios systematically decline below the initial capital. The combined effect of partial exposure and transaction costs results in an overall negative slope, with no strategy managing to remain above \$1,000 for any sustained period. However, the relative dispersion of the trajectories highlights that some approaches are better at slowing capital erosion than others.

Lower segment: 100% invested without costs There are temporary phases of performance (rising above \$1,000), but ultimately all curves fall back below the initial threshold. Increased volatility exacerbates both bullish and bearish extremes, confirming that no SVM strategy generated a positive net return over the period, despite occasional opportunities for gains.

Figure 9: Out-of-sample capital evolution of five SVM-based trading strategies from July 7, 2021, to February 24, 2022.



Note: The top panel shows results with 50% daily investment including 0.05% transaction costs, and the bottom panel shows results with 100% daily investment without transaction costs. The green and red shaded areas indicate performance above and below the initial capital baseline of \$1,000, respectively.

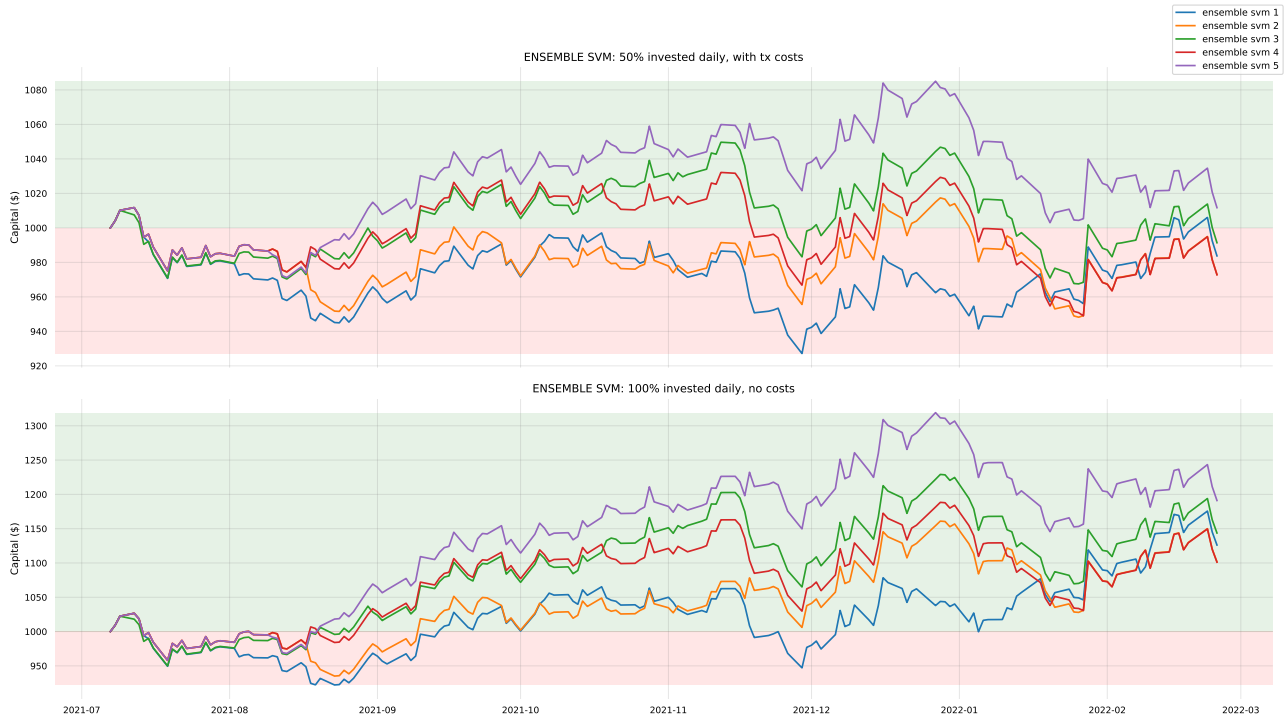
Sources: Historical AAPL stock data retrieved via `yfinance` (pypi.org/project/yfinance/), and Stocktwits messages dataset hosted on Kaggle (kaggle.com). Data covers the period from December 31, 2019 to February 27, 2022.

5.2.2 Ensemble SVM

Figure 10 illustrates the results of the two trading strategies over the period July 7, 2021 – February 24, 2022 for the Ensemble SVM model, for each of the five scenarios.

In the first configuration, partial exposure coupled with costs systematically leads to a decline in capital, with all trajectories falling back below the initial threshold despite a dispersion of around $100 between the best and worst performances. In contrast, the maximum leverage without fees allows each of the five executions in the lower panel to cross and sustainably maintain the $1,000 barrier, reaching peaks close to $1,300 for the **Ensemble SVM + RoBERTa** combination. Finally, the low dispersion of returns between each scenario shows stable performance for all five scenarios.

Figure 10: Out-of-sample capital evolution of five Ensemble SVM-based trading strategies from July 7, 2021, to February 24, 2022.



Note: The top panel shows results with 50% daily investment including 0.05% transaction costs, and the bottom panel shows results with 100% daily investment without transaction costs. The green and red shaded areas indicate performance above and below the initial capital baseline of $1,000, respectively.

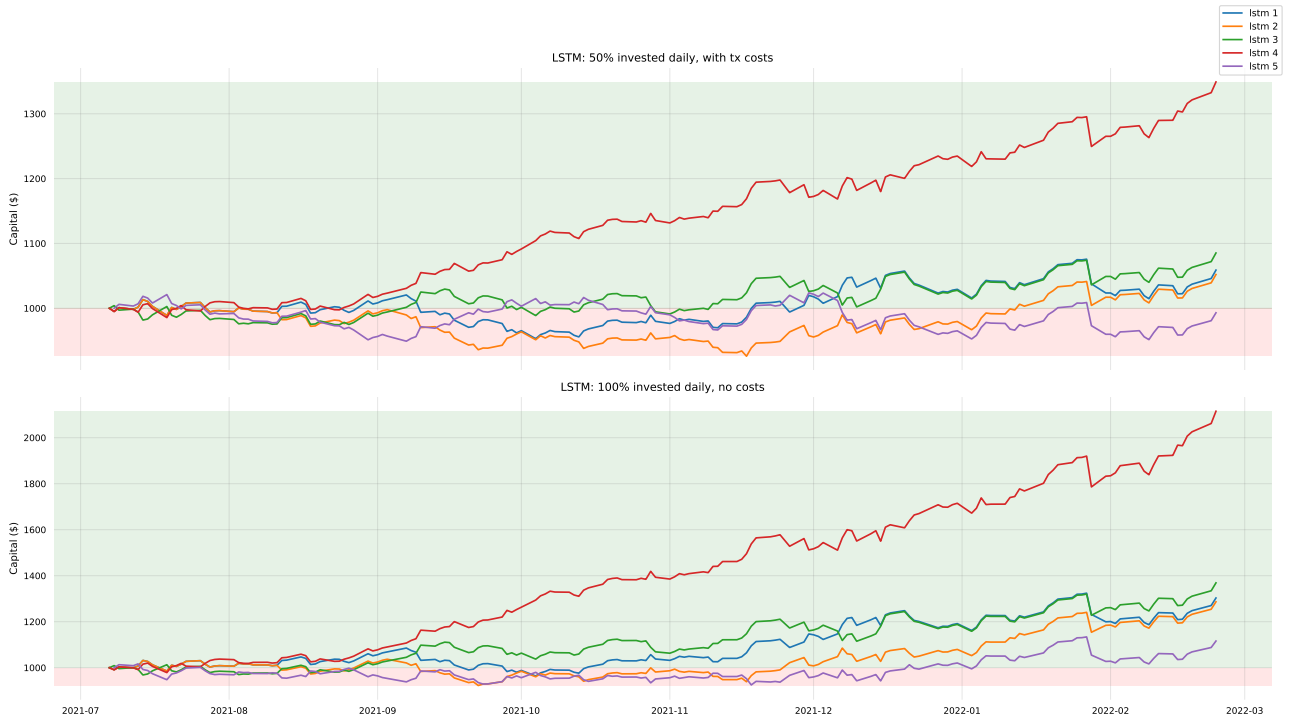
Sources: Historical AAPL stock data retrieved via `yfinance` (pypi.org/project/yfinance/), and Stocktwits messages dataset hosted on Kaggle (kaggle.com). Data covers the period from December 31, 2019 to February 27, 2022.

5.2.3 LSTM

Finally, Figure 11 illustrates the results of the two trading strategies over the period July 7, 2021 – February 24, 2022 for the LSTM model, for each of the five scenarios.

The results of the LSTM model over the out-of-sample period are the best among the three models tested. Despite transaction costs of 0.05% in the low-exposure trading strategy, four scenarios have a positive return, and scenario 5 is very close to the $1,000 mark.
In the second configuration (100% exposure with no transaction costs), all scenarios have a positive return, and the **LSTM + FinBERT** combination exceeds $2,000 in return in 161 days. This demonstrates the relevance of integrating variables from sentiment analysis to predict the direction of Apple's share price movement, and the value of sentiment analysis models trained on financial data.

Figure 11: Out-of-sample capital evolution of five LSTM-based trading strategies from July 7, 2021, to February 24, 2022.



Note: The top panel shows results with 50% daily investment including 0.05% transaction costs, and the bottom panel shows results with 100% daily investment without transaction costs. The green and red shaded areas indicate performance above and below the initial capital baseline of $1,000, respectively.

Sources: Historical AAPL stock data retrieved via `yfinance` (pypi.org/project/yfinance/), and Stocktwits messages dataset hosted on Kaggle (kaggle.com). Data covers the period from December 31, 2019 to February 27, 2022.

5.2.4 Comparison with the buy-and-hold strategy

Table 6 presents the results of the trading strategies for the fifteen models studied and compares them with the buy-and-hold strategy. The return of the passive "buy and hold" strategy over the period from July 7, 2021, to February 24, 2022, ranges from +11.19% to +13.05%, depending on the scenario (see column "Close Return (%)"). Of the fifteen models evaluated, six consistently outperform this benchmark:

- the four LSTM configurations in scenarios 1 to 4 (price only, auto-annotations, VADER, FinBERT),
- two variants of Ensemble SVM (scenarios 3 and 5).

In particular, the LSTM + FinBERT combination (scenario 4) produces a gain of +111.51% of capital, representing an outperformance of +100.31 points compared to buy & hold. LSTM + VADER (scenario 3) achieved +36.89%, or +25.70 points of outperformance, while LSTM in scenarios 1 and 2 posted +30.31% and +28.66%, respectively, both at least +17 points higher than the passive strategy. Conversely, simple SVMs failed to beat buy & hold, recording negative returns of between -11.14% and -39.68%, illustrating their limitations when faced with sequential and noisy data over the out-of-sample period studied.

Table 6: Summary statistics of the strategies' performance compared to passive holding from July 7, 2021, to February 24, 2022.

Strategy	Initial Capital	Final Capital	Capital Return (%)	Initial Close	Final Close	Close Return (%)	Outperformance (%)
LSTM (4)	1000.00	2115.06	111.51%	141.66	157.52	11.19%	100.31%
LSTM (3)	1000.00	1368.86	36.89%	141.66	157.52	11.19%	25.70%
LSTM (1)	1000.00	1303.14	30.31%	141.66	157.52	11.19%	19.12%
LSTM (2)	1000.00	1286.55	28.66%	141.66	157.52	11.19%	17.46%
Ensemble SVM (5)	1000.00	1191.03	19.10%	141.66	160.15	13.05%	6.06%
Ensemble SVM (3)	1000.00	1143.65	14.37%	141.66	160.15	13.05%	1.32%
Ensemble SVM (1)	1000.00	1126.14	12.61%	141.66	160.15	13.05%	-0.43%
LSTM (5)	1000.00	1115.97	11.60%	140.36	157.52	12.22%	-0.63%
Ensemble SVM (2)	1000.00	1101.59	10.16%	141.66	160.15	13.05%	-2.89%
Ensemble SVM (4)	1000.00	1101.20	10.12%	141.66	160.15	13.05%	-2.93%
SVM (1)	1000.00	888.59	-11.14%	141.66	160.15	13.05%	-24.19%
SVM (3)	1000.00	875.21	-12.48%	141.66	160.15	13.05%	-25.52%
SVM (2)	1000.00	785.37	-21.46%	141.66	160.15	13.05%	-34.51%
SVM (5)	1000.00	701.75	-29.83%	141.66	160.15	13.05%	-42.87%
SVM (4)	1000.00	603.22	-39.68%	141.66	160.15	13.05%	-52.72%

Sources: Historical AAPL stock data retrieved via `yfinance` (pypi.org/project/yfinance/), and Stocktwits messages dataset hosted on Kaggle (kaggle.com). Data covers the period from December 31, 2019 to February 27, 2022.

6 Discussion

6.1 Comparative Model Performance

Our results confirm the superiority of models enriched with sentiment variables over purely technical models. Across all tested scenarios, LSTM achieved an average F1 score of 57.06%, compared to 55.84% for Ensemble SVM and 54.53% for simple SVM. This advantage can be explained by the ability of LSTMs to capture temporal dependencies and changing regimes in financial series, as previously demonstrated by Jin et al. (2020) for the S&P 500.

6.2 Differentiated Contribution of Sentiment Signals

Among the various sentiment analysis methods, VADER stands out by propelling LSTM to its best F1 score (59.91%) thanks to its sensitivity to the informal tone and emojis characteristic of StockTwits. In contrast, FinBERT only brings a moderate gain (+0.11 points) and RoBERTa fine-tuned on StockTwits slightly penalizes LSTM (53.90%), possibly due to instability related to overfitting on a relatively small dataset. The modest performance of this sentiment analysis model (lowest F1 score among the scenarios using sentiment scores) can be explained by the fact that the model was trained only on auto-annotated messages, which introduces a bias (50% of messages in our dataset were not annotated, for example).

6.3 Implications for Trading Strategies

The out-of-sample simulation reveals that only sentiment-enriched LSTM consistently outperforms the passive "buy & hold" strategy. In particular, LSTM + FinBERT nearly doubles the initial capital (+111.51% vs. +11.19% for buy & hold) over the period and outperforms by more than 100 points. LSTM + VADER achieved +36.89%, a lead of +25.70 points. The Ensemble SVM + RoBERTa and Ensemble SVM + VADER combinations also outperformed the passive strategy over the period.

Conversely, simple SVMs posted negative returns (up to –39.68%), demonstrating their limitations in the face of volatility and textual noise.

6.4 Limitations and Improvement Perspectives

Period and universe studied: the study is limited to Apple (AAPL) over 2020–2022; results may vary depending on the asset profile and market phases.

Technical constraints: Limited computational resources and training time did not allow us to fully optimize the hyperparameters and explore all the architectures considered, nor to use a walk-forward method for the LSTM.

Relative F1 scores: despite promising performance, our F1 scores remain lower than some studies in the literature (e.g. 65% F1 score by Liu et al. (2023)).

Daily view of the trading strategies: the trading strategies only have a daily view, limiting the capture of mid-term market dynamics. More sophisticated approaches, integrating fine time granularity and adaptive algorithms (e.g., real-time algorithmic trading), could improve responsiveness and overall performance.

7 Conclusion

Ultimately, integrating sentiment signals from StockTwits into models proves decisive in anticipating the daily direction of Apple's stock price: the LSTM architecture consistently outperforms SVMs, with an average F1 score of 57.06% and peaking at 59.91% when VADER scores are used, nearly three points better than a model based solely on prices. Beyond this statistical gain, the economic value is striking: over the out-of-sample period from July 7, 2021, to February 24, 2022, the LSTM strategy enhanced by FinBERT transforms $1,000 in capital into more than $2,115 (+111.5%), while a simple buy-and-hold strategy only gains 11.2%. These results confirm that the "temperature" of collective sentiment is a tangible predictive lever, but they remain limited to a single asset, a daily frequency, and the 2020-2022 period. Exploring other securities, intraday horizons, and more advanced architectures is therefore the next step in consolidating and extending the scope of this approach.

Appendix

The complete source code associated with this work is publicly available on GitHub at the following URL: https://github.com/gaeldatascience/apple-stock-prediction

References

Antweiler, W. and Frank, M. Z. (2004). Is All That Talk Just Noise? The Information Content of Internet Stock Message Boards. *The Journal of Finance*, 59(3):1259–1294.

Araci, D. (2019). FinBERT: Financial Sentiment Analysis with Pre-trained Language Models. arXiv:1908.10063 [cs].

Baker, M. and Wurgler, J. (2006). Investor Sentiment and the Cross-Section of Stock Returns. *The Journal of Finance*, 61(4):1645–1680.

Batra, R. and Daudpota, S. M. (2018). Integrating StockTwits with sentiment analysis for better prediction of stock price movement. In *2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, pages 1–5.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3):273–297.

Gupta, R. and Chen, M. (2020). Sentiment Analysis for Stock Price Prediction. In *2020 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, pages 213–218.

Hiew, J. Z. G., Huang, X., Mou, H., Li, D., Wu, Q., and Xu, Y. (2019). BERT-based Financial Sentiment Index and LSTM-based Stock Return Predictability.

Hu, Z., Zhu, J., and Tse, K. (2013). Stocks market prediction using Support Vector Machine. In *2013 6th International Conference on Information Management, Innovation Management and Industrial Engineering*, volume 2, pages 115–118. ISSN: 2155-1472.

Hutto, C. and Gilbert, E. (2014). VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text. *Proceedings of the International AAAI Conference on Web and Social Media*, 8(1):216–225.

Jin, Z., Yang, Y., and Liu, Y. (2020). Stock closing price prediction based on sentiment analysis and LSTM. *Neural Computing and Applications*, 32(13):9713–9729.

Ko, C.-R. and Chang, H.-T. (2021). LSTM-based sentiment analysis for stock price forecast. *PeerJ Computer Science*, 7:e408.

Koukaras, P., Nousi, C., and Tjortjis, C. (2022). Stock Market Prediction Using Microblogging Sentiment Analysis and Machine Learning. *Telecom*, 3(2):358–378.

Liu, J.-X., Leu, J.-S., and Holst, S. (2023). Stock price movement prediction based on Stocktwits investor sentiment using FinBERT and ensemble SVM. *PeerJ Computer Science*, 9:e1403.

Loughran, T. and Mcdonald, B. (2011). When Is a Liability Not a Liability? Textual Analysis, Dictionaries, and 10-Ks. *The Journal of Finance*, 66(1):35–65.

Ren, R., Wu, D. D., and Liu, T. (2019). Forecasting Stock Market Movement Direction Using Sentiment Analysis and Support Vector Machine. *IEEE Systems Journal*, 13(1):760–770.

Sun, A., Lachanski, M., and Fabozzi, F. J. (2016). Trade the tweet: Social media text mining and sparse matrix factorization for stock market prediction. *International Review of Financial Analysis*, 48:272–281.