Stock Price Forecasting through Financial Headline Sentiment Analysis

Reuben Billian
Rutgers University
rib30@scarletmail.rutgers.edu

May 11, 2025

Abstract

I present a neural network-based framework for short-term stock price forecasting that combines traditional financial indicators with sentiment analysis derived from financial news headlines. My feature set includes sector weightings, transformer-based sentiment scores (DistilRoBERTa, FinBERT), rule-based sentiment metrics (VADER), price and volume data, momentum indicators, and news-driven volatility proxies for major firms. I train a Multi-Layer Perceptron (MLP) Regressor and benchmark its performance against a Random Forest Classifier to assess both regression accuracy and the model's ability to predict directional price movements. Results suggest that integrating textual sentiment features with market data improves predictive performance for short-term trend forecasting.

1 Introduction

Stock price prediction is a long-standing challenge in computational finance. Traditional approaches often incorporate a mix of technical indicators and fundamental data, but recent work suggests that unstructured textual information—such as financial news headlines and social media commentary—may also provide valuable predictive signals. As online content becomes more integrated into market behavior, sentiment-driven responses can impact short-term price movements.

Advancements in natural language processing, particularly transformer-based models, have improved the ability to quantify sentiment from financial text. In this paper, I examine whether incorporating sentiment scores from financial news headlines can enhance directional price trend classification. I train a multilayer perceptron (MLP) using both market data and sentiment features and compare its trend prediction performance against a Random Forest classifier (RFC).

2 Related Work

The integration of sentiment analysis into financial forecasting has been widely explored. Early approaches frequently used lexicon-based models such as VADER, TextBlob, and Flair to extract sentiment from financial news and social media. These models demonstrated that even simple heuristics can capture market-relevant signals and improve short-term forecasting performance [Junaid Maqboola(2023)].

More recently, transformer-based models have emerged as state-of-the-art in sentiment extraction. FinBERT, a BERT variant fine-tuned on financial text, offers strong performance on domain-specific tasks, while DistilRoBERTa provides a faster, more general-purpose alternative. According to [Ayesha Khaliq(2025)], FinBERT's domain adaptation leads to high accuracy on financial sentiment tasks, whereas DistilRoBERTa offers better efficiency and flexibility for real-time applications. Both have been shown to be effective, with the choice depending on task constraints such as latency and domain specificity.

Some studies have also applied recurrent models like Long Short-Term Memory (LSTM) networks to combine sentiment and historical stock data. While LSTMs can model temporal dependencies well, they tend to overfit when data is limited or noisy, which is often the case in financial settings [Adil MOGHARa*(2020)]. This motivates the use of simpler architectures, such as multilayer perceptrons (MLPs), which can still capture nonlinear relationships without the same degree of overfitting risk.

3 Data and Preprocessing

3.1 Sources

This study combines historical market data with financial news headline data over the period from 2010 to 2020. Market data were obtained from Investing.com. For SPY, used as a proxy for the S&P 500 index, I use daily Open, Close, and Volume data. For four major large-cap stocks—Apple (AAPL), ExxonMobil (XOM), Alphabet (GOOGL), and Microsoft (MSFT)—I include daily trading volume and percentage price change.

Financial text data were sourced from a Kaggle dataset containing approximately 1.4 million financial news headlines. These headlines were preprocessed and analyzed using three sentiment analysis models: FinBERT (domain-specific), DistilRoBERTa (general-purpose), and VADER (rule-based). Daily sentiment scores were computed by aggregating headline-level outputs and merged with the corresponding daily market data to form the final modeling dataset.

3.2 Data Preprocessing and Feature Engineering

The dataset was first sorted chronologically to ensure time consistency. The financial news dataset spanned from 2009-04-27 to 2020-06-11 and had significantly finer temporal granularity, with timestamps recorded down to the minute. In contrast, the most granular market data available for the stocks of interest was on a daily basis. As a result, all data had to be aggregated to the daily level for alignment.

To begin preprocessing the financial headlines, a secondary dataset containing historical sector weightings for the S&P 500 was used. Each headline was associated with the sector of its corresponding stock, and a new feature was computed to reflect the sector weighting of that stock within the index. This served as a useful proxy for the potential impact of news about individual companies on the overall SPY index, since it captures the relative importance of a stock within the broader market.

Next, sentiment scores were computed for each headline using three distinct sentiment analysis models: FinBERT, DistilRoBERTa, and VADER. FinBERT and DistilRoBERTa each generated a single sentiment score per headline, while VADER produced four scores: negative, neutral, positive, and compound. These scores, along with the sector weightings, were then aggregated by day using the mean, standard deviation, and count of all headlines appearing on that date. This daily aggregation allowed the textual data to be merged with daily price data for further modeling.

The market data was then parsed, cleaned, and reformatted. For SPY, daily Open, Close, and Volume data were retained. For Apple (AAPL), ExxonMobil (XOM), Alphabet (GOOGL), and Microsoft (MSFT), daily trading volume and percentage change in price were used. All market and sentiment data were merged into a single dataset aligned by date.

The final feature set included sentiment statistics, sector weighting metrics, price data, volume metrics, and a momentum indicator. The momentum feature was calculated as the difference between the current day's closing price and the previous day's closing price. The binary classification label was constructed by computing the difference between the SPY price at time $t + 3$ and at time t. If the difference was positive, the label was set to 1 (indicating an upward trend in the next three days); otherwise, it was set to 0. The complete list of features is as follows:

```
['Sector Weighting_mean', 'Sector Weighting_std', 'Sector Weighting_count',
 'DistilRoBERTa Scores Headline_mean', 'DistilRoBERTa Scores Headline_std',
 'DistilRoBERTa Scores Headline_count', 'FinBERT Scores Headline_mean',
 'FinBERT Scores Headline_std', 'FinBERT Scores Headline_count',
 'vader_neg_mean', 'vader_neg_std', 'vader_neg_count', 'vader_neu_mean',
 'vader_neu_std', 'vader_neu_count', 'vader_pos_mean', 'vader_pos_std',
 'vader_pos_count', 'vader_compound_mean', 'vader_compound_std',
 'vader_compound_count', 'Price', 'Open', 'Vol.', 'Apple Vol',
 'Change Apple %', 'Alphabet Vol', 'Change Alphabet %', 'Exxon Vol',
 'Change Exxon %', 'Microsoft Vol', 'Change Microsoft %', 'Momentum',
 'label']
```

4 Model

To evaluate the effectiveness of sentiment-enhanced features in predicting short-term stock price trends, I experimented with two distinct model architectures: a multilayer perceptron (MLP) classifier and a Random Forest classifier. Given the

limited size of the dataset (2633 rows × 34 columns), model complexity and overfitting were important considerations.

4.1 Multilayer Perceptron (MLP)

The MLP model was chosen as a simple yet flexible non-linear classifier. Unlike recurrent architectures such as LSTMs, which are prone to overfitting on small datasets, MLPs can capture non-linear relationships without explicitly modeling temporal sequences. However, neural networks are often considered black boxes, and interpreting their learned representations is challenging. This made evaluation on a held-out test set crucial for validating performance.

I used the following configuration for the final sentiment-augmented model:

```
MLPClassifier(
    hidden_layer_sizes=(200, 300),
    activation='relu',
    solver='adam',
    alpha=0.01,
    batch_size='auto',
    learning_rate='constant',
    max_iter=10000,
    random_state=42
)
```

To establish a baseline without sentiment features, a smaller version of the MLP was trained using only technical and market data:

```
MLPClassifier(
    hidden_layer_sizes=(100, 150),
    activation='relu',
    solver='adam',
    alpha=0.01,
    batch_size='auto',
    learning_rate='constant',
    max_iter=10000,
    random_state=42
)
```

4.2 Random Forest Classifier

Random Forests were used both as a benchmark model and as a tool to assess feature importance. As an ensemble of decision trees, Random Forests are robust to overfitting, require minimal feature scaling, and are easier to interpret than neural networks. One key advantage is their ability to estimate the relative importance of input features, allowing us to evaluate the contribution of sentiment features to model performance.

The following configuration was used for both the baseline and sentiment-augmented Random Forest models:

```
RandomForestClassifier(
    n_estimators=100,
    max_depth=None,
    class_weight='balanced',
    random_state=42
)
```

4.3 Evaluation Strategy

Each model was trained on the dataset twice—once without sentiment features and once with sentiment features—to assess their marginal contribution. The models were evaluated on a time-based 90/10 train-test split without shuffling, to preserve the temporal ordering of financial data and avoid data leakage. Classification performance was measured

using accuracy, precision, recall, and F1 score on the held-out test set. The training set consisted of 90% of the data, and 10% was reserved for testing.

All models were implemented using standard Python libraries, including `scikit-learn`, `pandas`, `numpy`, `matplotlib`, `seaborn`, and `scipy`. While most features were left unscaled for compatibility with tree-based models, volume-related features—such as SPY volume and the trading volumes of individual stocks—were scaled to account for their large magnitude (often in the millions). The primary optimization objective across all experiments was to maximize trend classification accuracy on the test set.

5 Evaluation

5.1 Baseline Classification Performance (Without Sentiment)

To establish a baseline, I first trained the MLP and Random Forest classifiers using only technical and market data, excluding all sentiment-based features.

MLP (Baseline)

Metric	Value
Accuracy	43.6%
Precision	52.2%
Recall	43.6%
F1 Score	42.2%

Table 1: MLP Classification metrics on test set (baseline, without sentiment).

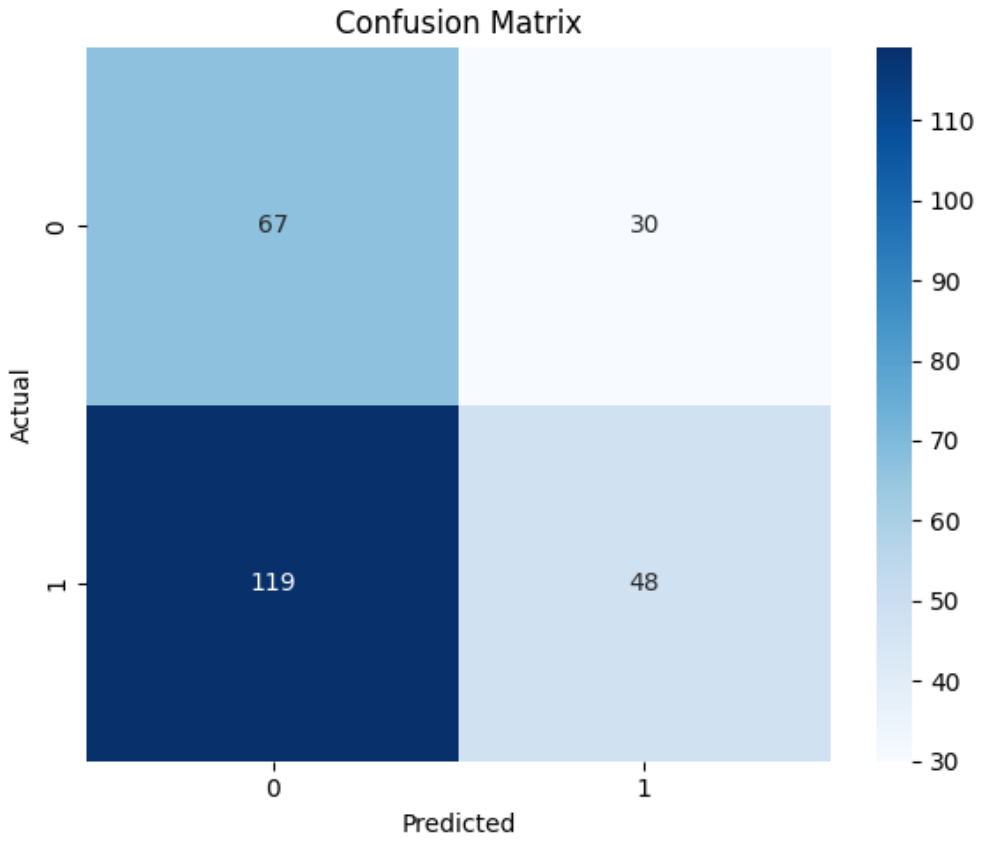

Figure 1: Confusion matrix for MLP baseline classifier (no sentiment features).

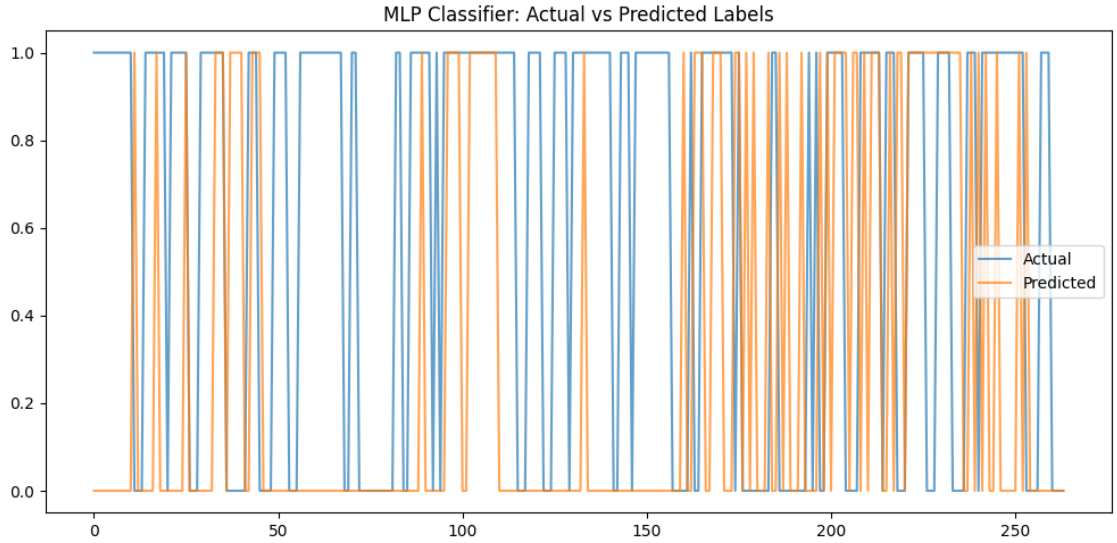

Figure 2: MLP baseline model: Actual vs. Predicted labels on test set.

RFC (Baseline)

Metric	Value
Accuracy	51.1%
Precision	64.2%
Recall	51.5%
F1 Score	57.1%

Table 2: Random Forest metrics on test set (baseline, without sentiment).

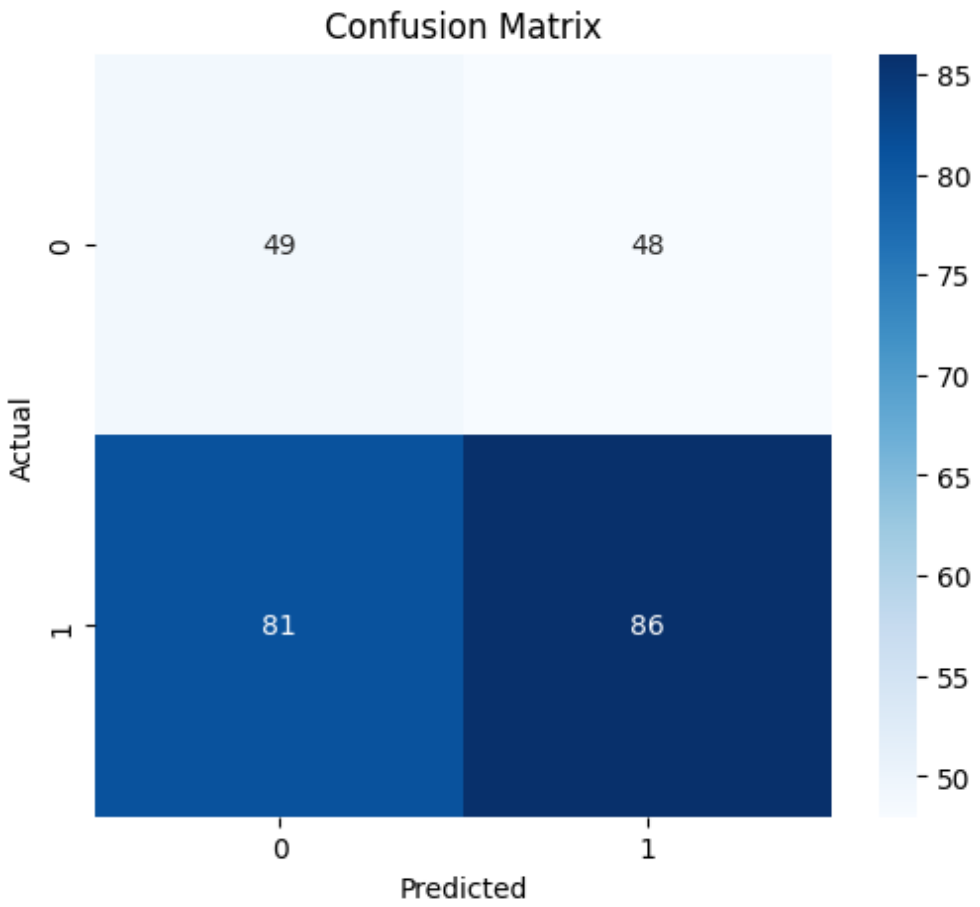

Figure 3: Confusion matrix for RFC baseline classifier (no sentiment features).

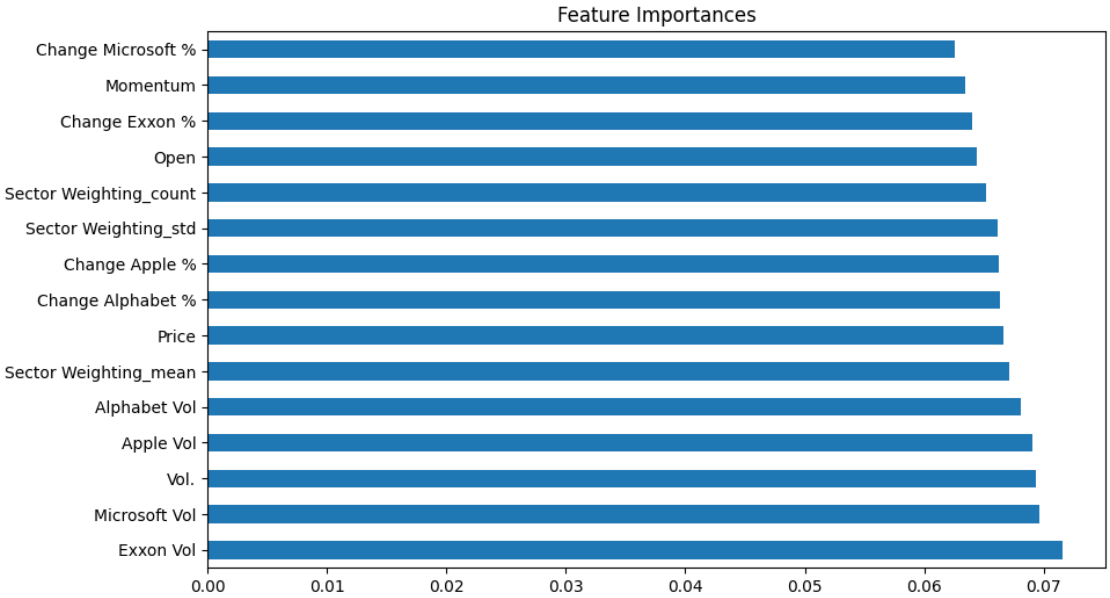

Figure 4: Feature importance for Random Forest classifier (baseline, no sentiment features).

5.2 Performance with Sentiment Features

Next, I evaluated both classifiers using the full feature set, including sentiment scores derived from financial news headlines. These features included outputs from FinBERT, DistilRoBERTa, and VADER, aggregated daily.

MLP (With Sentiment)

Metric	Value
Accuracy	54.9%
Precision	54.4%
Recall	54.9%
F1 Score	54.7%

Table 3: MLP metrics on test set (with sentiment features).

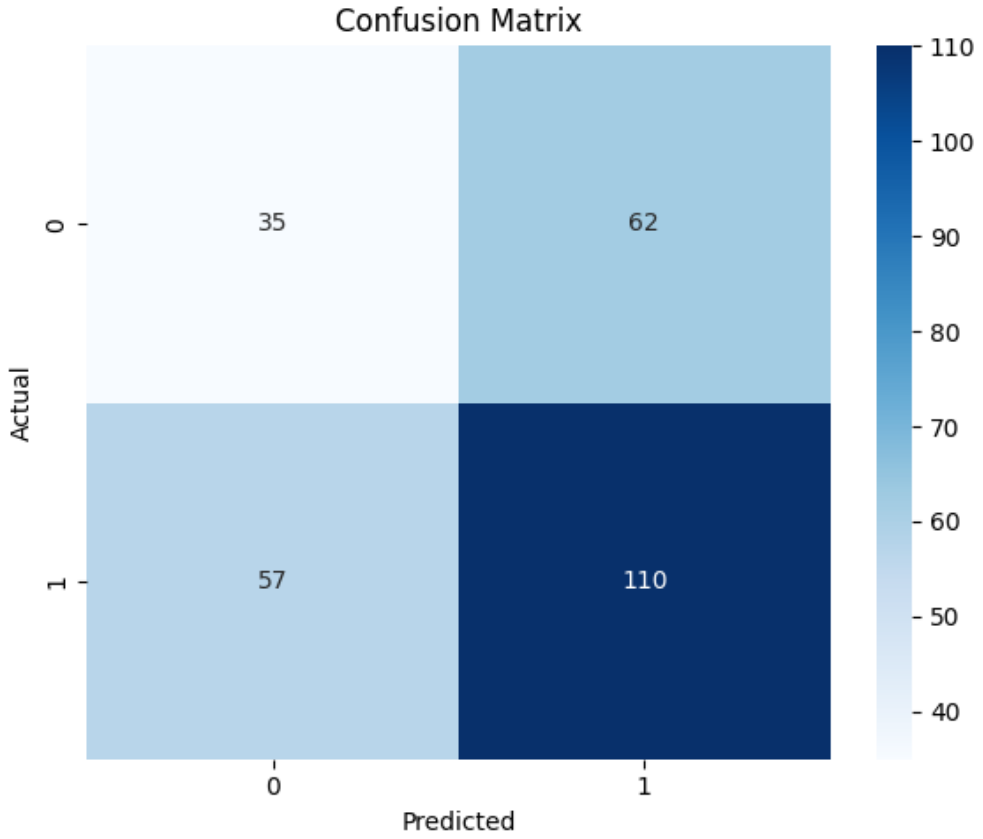

Figure 5: Confusion matrix for MLP classifier (with sentiment features).

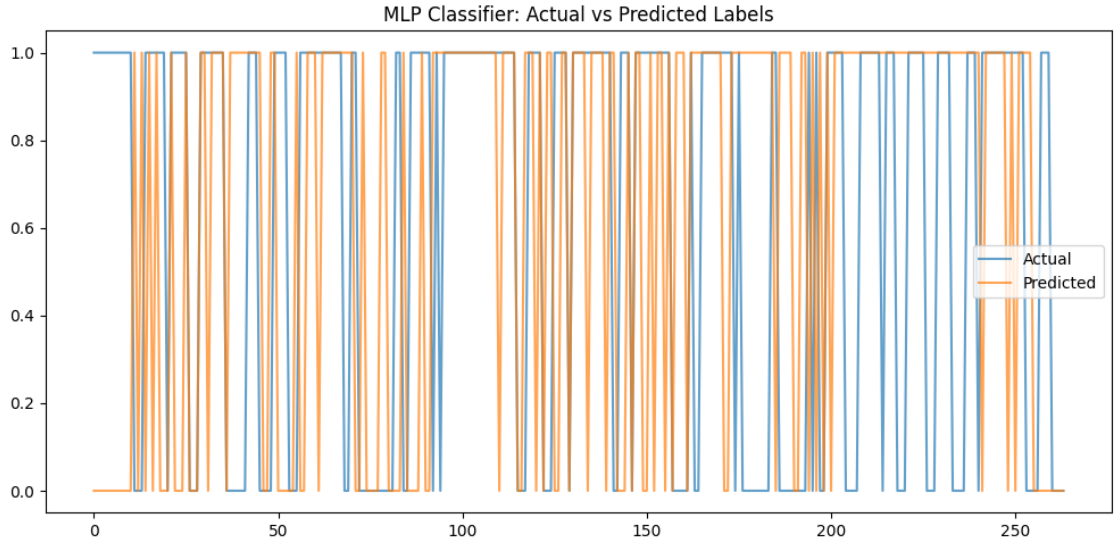

Figure 6: MLP with sentiment: Actual vs. Predicted labels on test set.

RFC (With Sentiment)

Metric	Value
Accuracy	62.9%
Precision	67.2%
Recall	80.8%
F1 Score	73.4%

Table 4: Random Forest metrics on test set (with sentiment features).

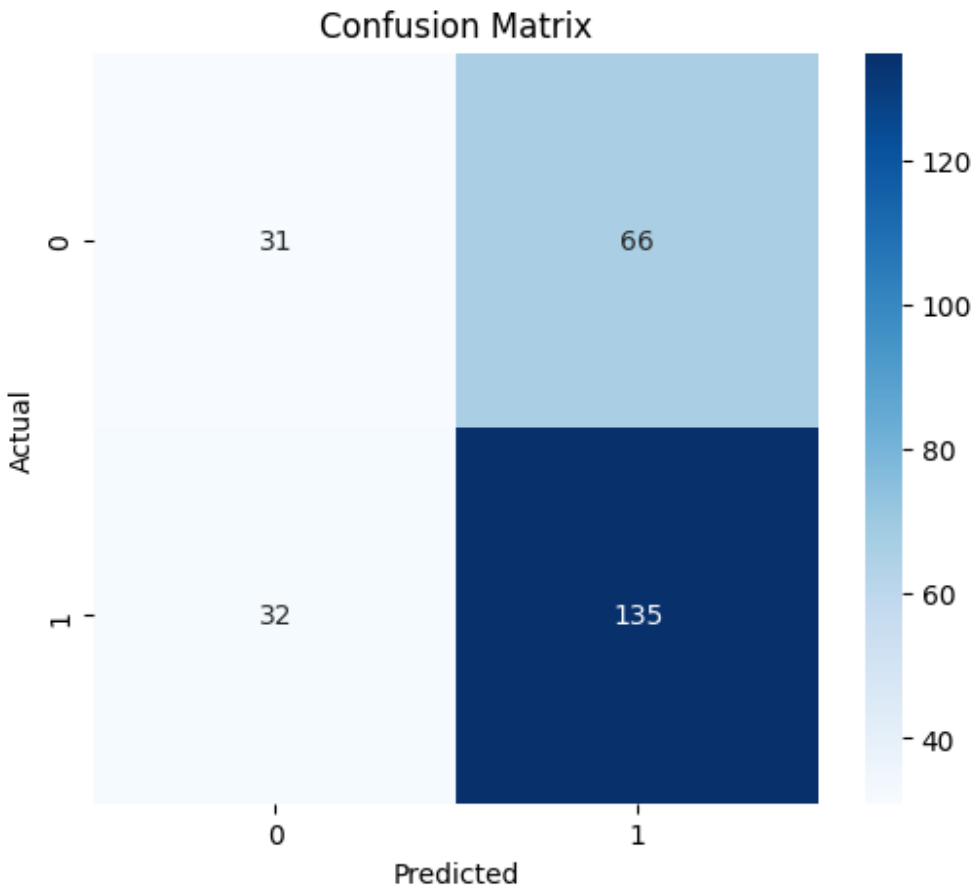

Figure 7: Confusion matrix for RFC classifier (with sentiment features).

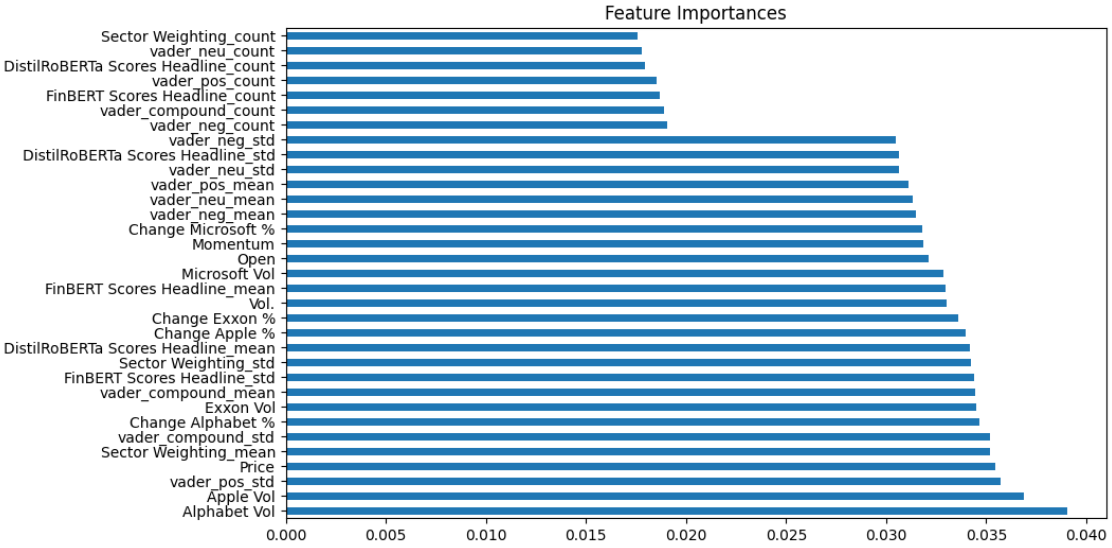

Figure 8: Feature importance for Random Forest classifier (with sentiment features).

5.3 Summary and Observations

The inclusion of sentiment features led to clear performance improvements in both models. The Random Forest classifier showed the strongest results overall, achieving a 73.4% F1 score and 62.9% accuracy with sentiment features. The MLP also benefited, increasing its accuracy from 43.6% to 54.9%. These results suggest that daily sentiment

extracted from financial headlines can contribute meaningfully to predicting short-term market direction. The feature importance plots also help highlight which variables—both market-based and sentiment-based—contributed most to classification performance.

6 Discussion

Feature-importance analysis (Figure 8) shows that traditional market variables—such as Alphabet volume, Apple volume, and closing price—remain the strongest predictors, while sentiment-derived features (VADER, FinBERT, DistilRoBERTa scores and their aggregates) occupy the middle ranks. This distribution of importance underscores the growing value of integrating alternative textual information—such as real-time social media feeds and financial news articles—into predictive systems, complementing traditional price and volume indicators.

Between the two architectures, the Random Forest consistently outperformed the MLP, likely because its bagging ensemble is more robust to overfitting on the limited dataset (approximately 2600 daily observations) and less sensitive to feature scaling. Additionally, tree-based models provide intuitive feature-importance estimates, clarifying which variables—both market-based and sentiment-based—drive model decisions.

From a practical standpoint, a roughly 10% boost in trend-classification accuracy can materially enhance trading signals—improving entry/exit timing and risk management—though real-world deployment must account for transaction costs, slippage, and inference latency.

Looking ahead, promising directions include:

- **Temporal models:** LSTM or Transformer architectures to capture multi-day dependencies and sequence effects.
- **Richer features:** Incorporate intraday price and volume data, option-implied volatility, and expanded alternative sources such as Twitter, Reddit, or YouTube comments alongside news headlines.
- **Ensembles:** Hybrid stacking of tree-based and neural models, or calibrated confidence estimates for more robust predictions.
- **Expanded sentiment coverage:** Fuse headline sentiment with streaming social media sentiment to capture rapid shifts in market mood.

7 Conclusion

I presented a hybrid model that combines market indicators with sentiment features, yielding an approximate 10% gain in short-term trend classification accuracy. Feature-importance analysis confirms that sentiment adds valuable signal alongside core price and volume metrics. Capturing directional trends remains challenging without temporal context, motivating the use of sequence models. Future work will explore time-series architectures, longer sentiment windows, and real-time deployment for more robust forecasting.

References

[Adil MOGHARa*(2020)] Mhamed HAMICHE Adil MOGHARa*. Stock market prediction using lstm recurrent neural network. *Science Direct*, 2020.

[Ayesha Khaliq(2025)] Sophia Ajaz Fawad Hussain Paul Ayesha Khaliq, Asif Ali. Comparative analysis of finbert and distilroberta for nlp-based financial insights in pakistan's stock market. *Spectrum of Engineering Sciences*, 2025.

[Junaid Maqboola(2023)] Ravreet Kaura Ajay Mittala Ishfaq Ali Ganaieb Junaid Maqboola, Preeti Aggarwala*. Stock prediction by integrating sentiment scores of financial news and mlp-regressor: A machine learning approach. *Science Direct*, 2023.