Forecasting Apple Stock Direction

Using Time-Series Features and Sentiment Analysis

Dimitrios Loukas

June 2025

Abstract

In this project, we built a complete machine learning pipeline to predict whether the Apple (AAPL) stock price will go up or down. We started by analyzing the historical data to understand patterns in price, volatility, and seasonality. Based on these insights, we created several new features, including lagged returns, technical indicators like moving averages and RSI, calendar effects, and sentiment scores from financial news.

To reduce complexity and keep only the most useful information, we used methods like correlation analysis, Random Forest importance, and SHAP values to select the most relevant features.

We tested different types of models, from simple ones like Logistic Regression and Random Forest to more advanced ones like XGBoost and LSTM. For each model, we compared how it performs using a small set of selected features versus the full enriched dataset. We also tested how much sentiment information helps improve prediction, and tried an attention-based LSTM to focus on important time steps.

The models were evaluated using metrics like accuracy, F1-score, AUC, and their harmonic mean. While the task is difficult and no model performed exceptionally well, models that included sentiment and used more features tended to perform slightly better in terms of AUC.

Contents

1 Introduction

This project focuses on predicting the short-term directional movement of Apple Inc. (AAPL) stock. The analysis integrates time-series features (such as lagged returns, volatility, and momentum indicators) with sentiment data obtained from news articles related to the company. The sentiment signals are computed on a daily basis and matched to the corresponding stock data. Multiple predictive models are used and compared, including baseline classifiers, gradient boosting, and deep learning models based on LSTM. Each model is evaluated on its ability to detect positive movements in the stock over a short horizon, based on different feature combinations.

The purpose of this study is to explore whether sentiment-based signals can add value to traditional price-based features when modeling stock behavior. This type of modeling can be useful for analysts who seek to understand short-term stock behavior, or for exploring the integration of alternative data sources in financial forecasting. However, stock price prediction remains inherently challenging due to market noise, external events, and human-driven sentiment. As such, the results should be interpreted with care, and the models evaluated in context.

2 Data Collection and Preprocessing

The dataset consists of historical stock price data for Apple Inc. (AAPL), downloaded using the Yahoo Finance API. The time range spans from 2010 to 2024 and includes standard daily trading features such as Open, High, Low, Close, and Volume.

To enrich the feature space, several technical indicators were computed, including daily returns, lagged returns over multiple days, rolling volatility, and momentum indicators such as RSI and MACD. These features aim to capture price dynamics and short-term trends.

In addition, daily sentiment scores were extracted from financial news headlines related to Apple. The dataset includes sentiment polarity and the number of news articles per day. The sentiment information was aggregated by date and merged with the technical dataset based on matching dates. The complete set of features is described in more detail in Section 4.

Missing values resulting from rolling calculations and data alignment were removed. The final dataset was saved in enriched form and used throughout the modeling pipeline.

The final dataset combines price-based features, technical indicators, and sentiment data. The technical features were computed from historical price movements, while the sentiment scores were derived from financial news headlines.

3 Exploratory Data Analysis

Before starting any modeling or feature design, it is important to understand the structure and behavior of the raw data. Exploratory data analysis (EDA) helps assess the quality of the dataset, detect patterns or irregularities, and guide feature engineering.

The dataset consists of daily trading records for Apple Inc. from 2010 to 2024, including open, high, low, close prices and traded volume. Additional time-derived fields like year, month, and day of the week were extracted for seasonality analysis.

The plot in Figure 1 shows the evolution of the closing price. Apple's price shows long-term upward movement, but also exhibits significant short-term fluctuations, which justify the need for daily-level prediction models.

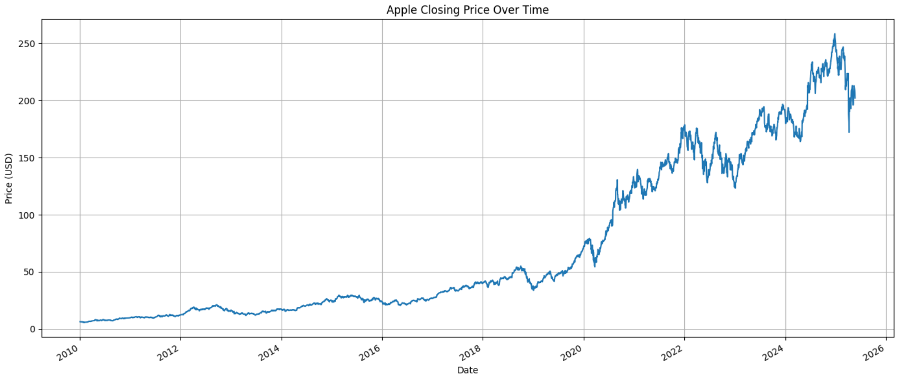

Figure 1: Daily closing price of Apple Inc. from 2010 to 2024.

We also examined seasonal behavior by aggregating price data per month and per year. As shown in Figure 2, monthly closing prices exhibit consistent seasonal variation, suggesting potential value in calendar-based features like `Month` and `DayOfWeek`.

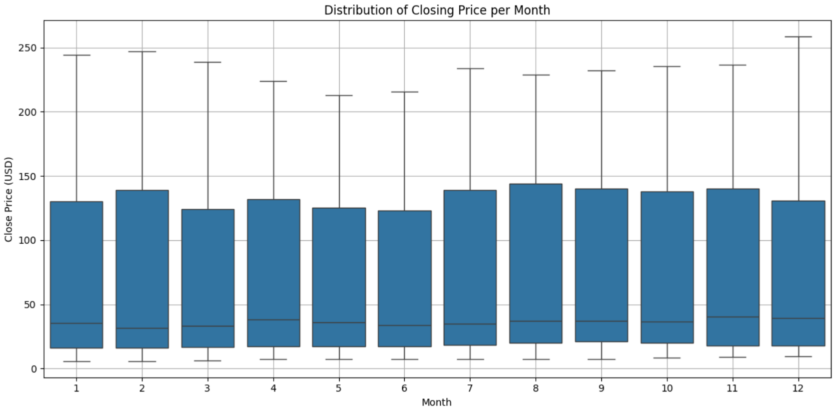

Figure 2: Monthly distribution of closing prices, showing seasonality.

Trading volume also carries important signals. Spikes in volume often coincide with market-moving events or announcements. Figure 3 shows daily trading volume over time, revealing both gradual trends and abrupt shifts.

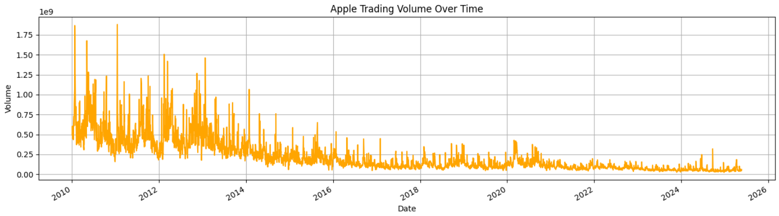

Figure 3: Daily trading volume over time. Peaks correspond to major market events.

To assess redundancy among numerical variables, we computed the correlation matrix Figure 4. Highly correlated features can distort model training and reduce generalizability. This analysis informed later steps such as feature pruning and selection.

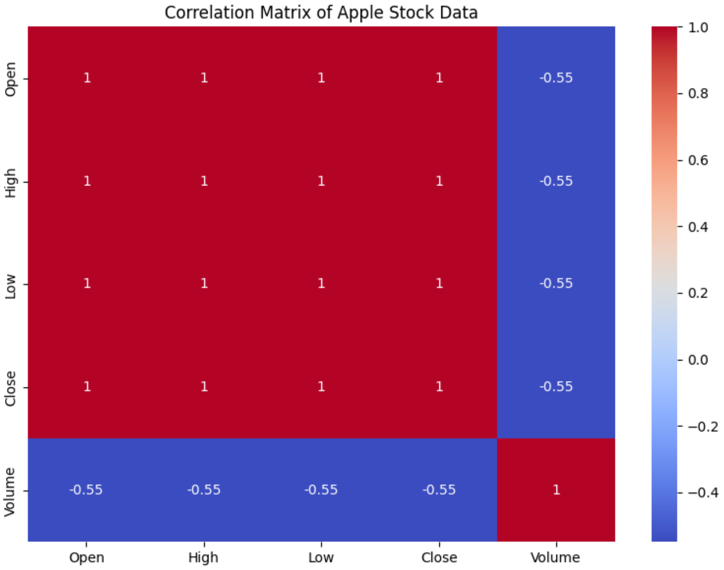

Figure 4: Correlation matrix of selected numeric features.

In summary, while EDA does not offer direct conclusions, it helped highlight some useful characteristics of the dataset. We observed patterns in seasonality, changes in volume, and price volatility, which may offer useful signals for prediction. These insights supported the decision to include short-term technical indicators (such as moving averages and lag returns) and calendar-based features in the modeling phase.

4 Feature Overview

The feature set was designed to combine different perspectives of stock behavior: internal market signals, statistical summaries, and external sentiment indicators. While some features capture recent price dynamics (e.g., lagged returns and rolling averages), others are designed to reflect market momentum or investor sentiment.

Rather than relying on a single category of inputs, the aim was to create a diverse representation of factors that might influence short-term stock movements. This includes trend indicators, volatility measures, trading activity, and aggregated daily sentiment from financial news.

Features were selected not only based on financial theory but also with practical constraints in mind. All of them are derived from publicly available data, are straightforward to compute, and are aligned on a daily basis, making them suitable for integration into a unified modeling framework.

The final selection avoids highly redundant or unstable variables and focuses on features that are both interpretable and likely to carry predictive signal. A complete list is shown in Table 1.

Table 1: Overview of all features with descriptions

Feature	**Description**
Daily_Return	Percentage change in closing price from previous day. Captures daily price movement.
Lag_1 to Lag_10	Daily returns from previous days. Provide short-term memory of recent trends.
Rolling_Return_5, Rolling_Return_10	Return over rolling 5-day or 10-day window. Capture short-term momentum.
Volatility_20, Volatility_50	Rolling standard deviation over 20 or 50 days. Capture recent market volatility.
MA_20, MA_50	Moving averages of closing price. Smooth short-term and medium-term price trends.
MA_Cross	Binary indicator: 1 if MA_20 > MA_50 (bullish signal), else 0.
Price_Above_MA_20, Price_Above_MA_50	Indicates whether current price is above respective MA.
RSI_14	Relative Strength Index over 14 days. Measures momentum and potential overbought/oversold conditions.
MACD, MACD_Signal	Momentum indicators based on the difference of EMAs. Used for trend identification.
Intraday_Range	Difference between daily high and low price. Proxy for daily volatility.
Volume_Avg_20	Average trading volume over the last 20 days.
Volume_Diff	Difference between current volume and Volume_Avg_20. Highlights unusual activity.
Month, DayOfWeek	Encodes potential seasonality and weekday patterns.
Sentiment_Mean	Average sentiment polarity score from daily Apple-related news.
Sentiment_Pos_Mean, Sentiment_Neg_Mean, Sentiment_Neu_Mean	Average positive, negative, and neutral sentiment scores respectively.
News_Count	Number of Apple-related news articles on a given day. Measures market attention.
Target, Target_3d, Target_5d	Binary classification targets: predict next-day or multi-day (3 or 5) movement over threshold.

5 Feature Importance Analysis

5.1 Why Feature Selection Matters

Feature selection is an important step in predictive modeling. Including too many variables can lead to overfitting, increased computational cost, and reduced interpretability. By identifying the most relevant features, models can focus on informative signals and generalize better to unseen data.

In this project, feature selection also helps distinguish which types of inputs (e.g., lagged returns, sentiment scores, volatility indicators) contribute most to the predictive task. It supports model transparency and helps reduce redundancy among correlated features.

5.2 Feature Importance Methods

To evaluate feature importance, three methods were used:

- **Pearson Correlation:** Measures how strongly each feature is related to the target. It looks at whether the values of a feature go up or down together with the target.
- **Random Forest Importances:** A tree-based method that shows how useful each feature is for making accurate predictions. It looks at how often a feature is used to split the data in the decision trees.
- **SHAP Values:** A method that shows how much each feature contributes to a model's prediction. It works by testing how the prediction changes when a feature is included or left out. SHAP helps understand which features push the prediction higher or lower.

Each method provides a complementary perspective on feature relevance.

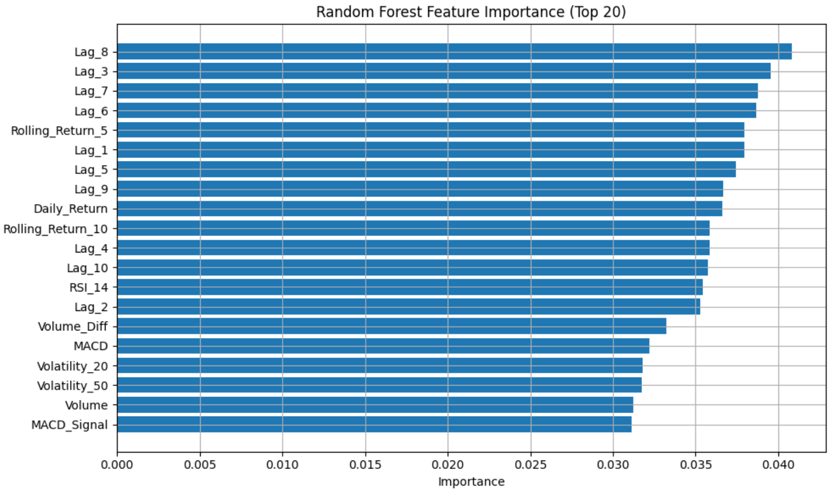

Figure 5: Random Forest feature importances (top-ranked features shown).

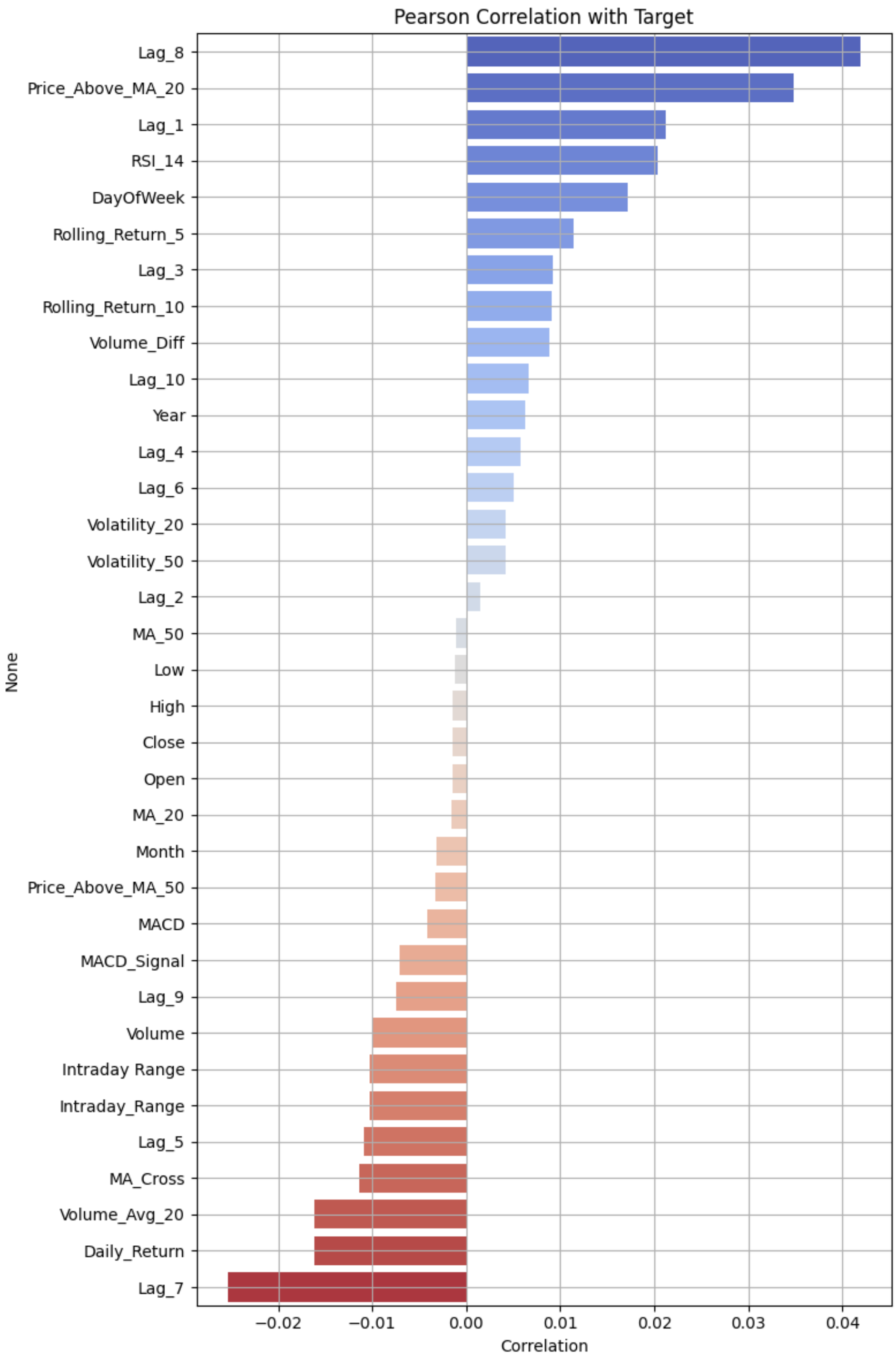

Figure 6: Pearson correlation between numeric features and the target variable.

5.3 Selection Criteria and Thresholds

To reduce the number of input features and keep only the most useful ones, a SHAP-based selection method was used. For each feature, the average of its absolute SHAP values across all samples was calculated. This number shows how much, on average, the feature affects the model's prediction — regardless of direction.

A threshold of 0.01 was applied, meaning that only features with an average impact greater than 1% were kept. This helped remove features with very low contribution that might add noise or redundancy.

In addition to the numeric threshold, the SHAP summary plot was reviewed visually. Features that showed a clear color pattern and consistent separation in values were preferred, as these are more likely to carry meaningful information for prediction.

To illustrate how SHAP values explain individual predictions, Figure 7 shows a single prediction and how each feature contributes to increasing or decreasing the output. Blue arrows push the prediction lower, while red arrows push it higher.

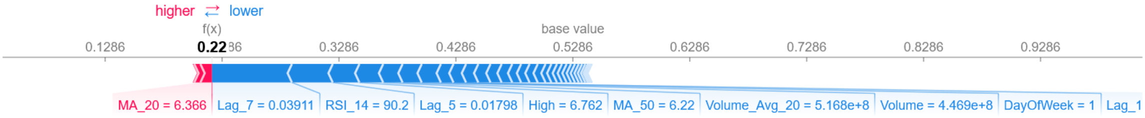

Figure 7: SHAP force plot showing how different features contribute to a single prediction.

The SHAP summary plot in Figure 8 shows the overall impact of each feature across all samples. Features are ranked by importance, and the color indicates the original value of the feature (e.g., red = high, blue = low). The horizontal spread shows how much each feature can change the model's prediction.

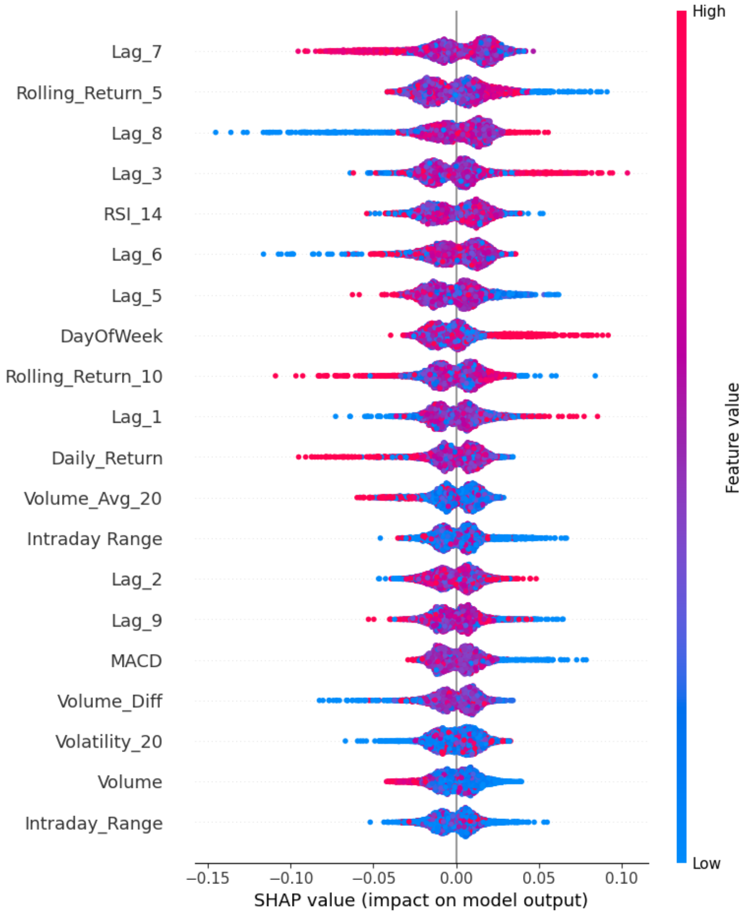

Figure 8: SHAP summary plot showing global feature importance and value distributions.

The final selected features used for baseline and initial modeling are: **Lag_7**, **Rolling_Return_5**, **Lag_8**, **Lag_6**, **RSI_14**, and **DayOfWeek**. These were chosen for their predictive signal and interpretability, based on both statistical scores and financial intuition.

In addition to the strict SHAP threshold of 0.01 used for selecting six core features, a relaxed threshold (around 0.005–0.008) was applied to construct a broader yet still focused feature set. These features were used in extended versions of MLP and XGBoost to assess whether additional information improves predictive power without introducing excessive noise.

5.4 Summary

Feature selection helped reduce the input space from over 40 variables to a manageable set of core features. The selected features span multiple categories: lagged price signals, momentum, calendar effects, and technical indicators. These were used in the first series of models (including logistic regression, random forest, MLP, and LSTM) to evaluate performance with a compact and interpretable input set.

To better understand the impact of additional variables, all models were also trained using the full set of enriched features. This allowed direct comparison between minimal and extended feature configurations. The overall modeling process and rationale are described in Section 6.

6 Implementation Workflow

This section outlines the modeling steps followed in the project, explaining how different feature sets were used and how models were evaluated across stages. The goal was to compare simple and more complex models using both compact and full feature sets, including sentiment information.

The modeling process was structured in the following stages:

1. **Baseline models using SHAP-selected features:**
   Logistic Regression and Random Forest were trained using the six features selected based on SHAP importance. These models served as a reference point for more advanced approaches.

2. **Neural network models:**
   A simple MLP (Multilayer Perceptron) was trained using the same SHAP-selected features to assess performance improvements from non-linear modeling. A second version of the MLP was trained using an intermediate subset of 10 features that passed a relaxed SHAP threshold. This allowed testing whether slightly more information (but still below full feature complexity) could lead to better generalization.

3. **XGBoost and LSTM using SHAP-selected features:**
   Two more powerful models were evaluated next. XGBoost was tuned with grid search, and a custom LSTM network was trained using the SHAP feature set. This stage allowed comparisons between tree-based and sequential deep learning models on the same compact input.

4. **XGBoost and LSTM using all engineered features:**
   To investigate whether additional features could improve prediction, both models were retrained using the complete set of technical and sentiment features. For XGBoost, different forecasting horizons (3-day and 5-day targets) were also explored.

5. **Sentiment-enhanced LSTM models:**
   LSTM was extended further by incorporating sentiment signals from financial news. Two variations were tested: one standard LSTM using sentiment-enriched features, and one with an attention mechanism to focus on the most influential parts of the input sequence.

6. **XGBoost with full sentiment features:**
   Finally, an XGBoost model was trained using the complete feature set including sentiment. This allowed fair comparison with its LSTM counterpart. “‘

Table 2 summarizes the models and the corresponding feature sets used during experimentation.

Model	**Feature Set Used**
Logistic Regression	SHAP-selected (6)
Random Forest	SHAP-selected (6)
MLP	SHAP-selected (6)
MLP	SHAP-intermediate (10)
XGBoost	SHAP-selected (6)
XGBoost (Target 3 days)	SHAP-intermediate (10)
XGBoost (Target 3 days)	All features
XGBoost (Target 5 days)	All features
XGBoost + Sentiment	All features
LSTM	SHAP-selected (6)
LSTM	All features
LSTM + Sentiment	All features
LSTM + Sentiment + Attention	All features

Table 2: Overview of models and feature sets used.

7 Results and Evaluation

To compare the performance of the different models, a consistent evaluation procedure was followed across all experiments. Each model was trained on the same training set and evaluated on a separate test set to ensure comparability.

The primary evaluation metrics used were accuracy, F1-score (weighted), and ROC AUC. Accuracy provides a general measure of correct predictions, while F1-score accounts for class imbalance by balancing precision and recall. The AUC metric reflects the model's ability to separate the two classes across thresholds, offering an additional perspective on classification performance.

To better summarize overall model performance, the harmonic mean of F1-score and AUC was also computed. This combined score balances classification quality with discrimination power and was used as a single reference point when comparing models side by side.

The harmonic mean is calculated using the formula:

$$\text{Harmonic Mean} = \frac{2 \cdot \text{F1} \cdot \text{AUC}}{\text{F1} + \text{AUC}} \tag{1}$$

This type of average gives more weight to the lower of the two values. It was chosen to ensure that a model performs reasonably well in both classification and ranking tasks, rather than excelling in only one.

For each model, predictions on the test set were used to calculate all metrics. In cases where probabilistic outputs were available (e.g., from logistic regression, XGBoost, and neural networks), ROC curves were also generated to visualize the tradeoff between true positive and false positive rates.

All results were collected and reported in a unified table. The same target variable was used across models unless otherwise noted.

7.1 Performance Summary

Table 3 summarizes the evaluation metrics for all models tested, sorted by the harmonic mean of F1 and AUC. All models were tested on the same test set, ensuring fair comparison.

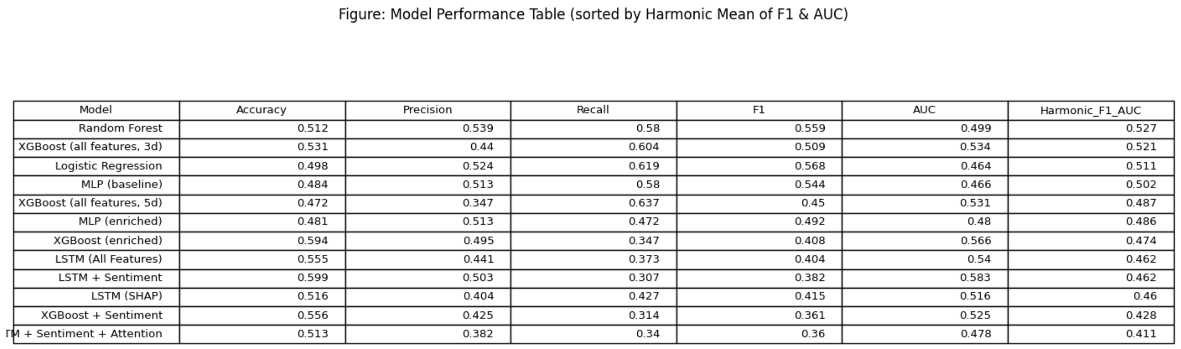

Figure: Model Performance Table (sorted by Harmonic Mean of F1 & AUC)

Model	Accuracy	Precision	Recall	F1	AUC	Harmonic_F1_AUC
Random Forest	0.512	0.539	0.58	0.559	0.499	0.527
XGBoost (all features, 3d)	0.531	0.44	0.604	0.509	0.534	0.521
Logistic Regression	0.498	0.524	0.619	0.568	0.464	0.511
MLP (baseline)	0.484	0.513	0.58	0.544	0.466	0.502
XGBoost (all features, 5d)	0.472	0.347	0.637	0.45	0.531	0.487
MLP (enriched)	0.481	0.513	0.472	0.492	0.48	0.486
XGBoost (enriched)	0.594	0.495	0.347	0.408	0.566	0.474
LSTM (All Features)	0.555	0.441	0.373	0.404	0.54	0.462
LSTM + Sentiment	0.599	0.503	0.307	0.382	0.583	0.462
LSTM (SHAP)	0.516	0.404	0.427	0.415	0.516	0.46
XGBoost + Sentiment	0.556	0.425	0.314	0.361	0.525	0.428
TM + Sentiment + Attention	0.513	0.382	0.34	0.36	0.478	0.411

Table 3: Model performance summary (sorted by Harmonic Mean of F1 and AUC).

7.2 Discussion

From the table, we observe that Random Forest surprisingly achieved the highest harmonic mean, despite being a simple model. However, its lead is marginal and may stem from its balanced precision and recall.

The XGBoost model with all features (3d horizon) performs competitively, likely benefiting from richer information and a shorter prediction window. Its strong recall drives the high F1.

Traditional models like Logistic Regression and MLP baseline also showed decent generalization with simple SHAP-selected features, making them good starting points for lightweight forecasting.

Among deep learning approaches, LSTM with sentiment achieved good AUC (0.583), but had lower recall, which hurt its F1. Interestingly, LSTM + attention, despite its added complexity, underperformed both in AUC and harmonic mean — possibly due to overfitting or noise sensitivity.

7.3 ROC Curve Analysis

To better understand discrimination ability, we also plotted the ROC curves across models. Despite their limitations in imbalanced or low-signal settings, ROC curves offer a helpful view of each model's ability to rank positive cases higher than negatives. This view complements threshold-dependent metrics like accuracy or F1.

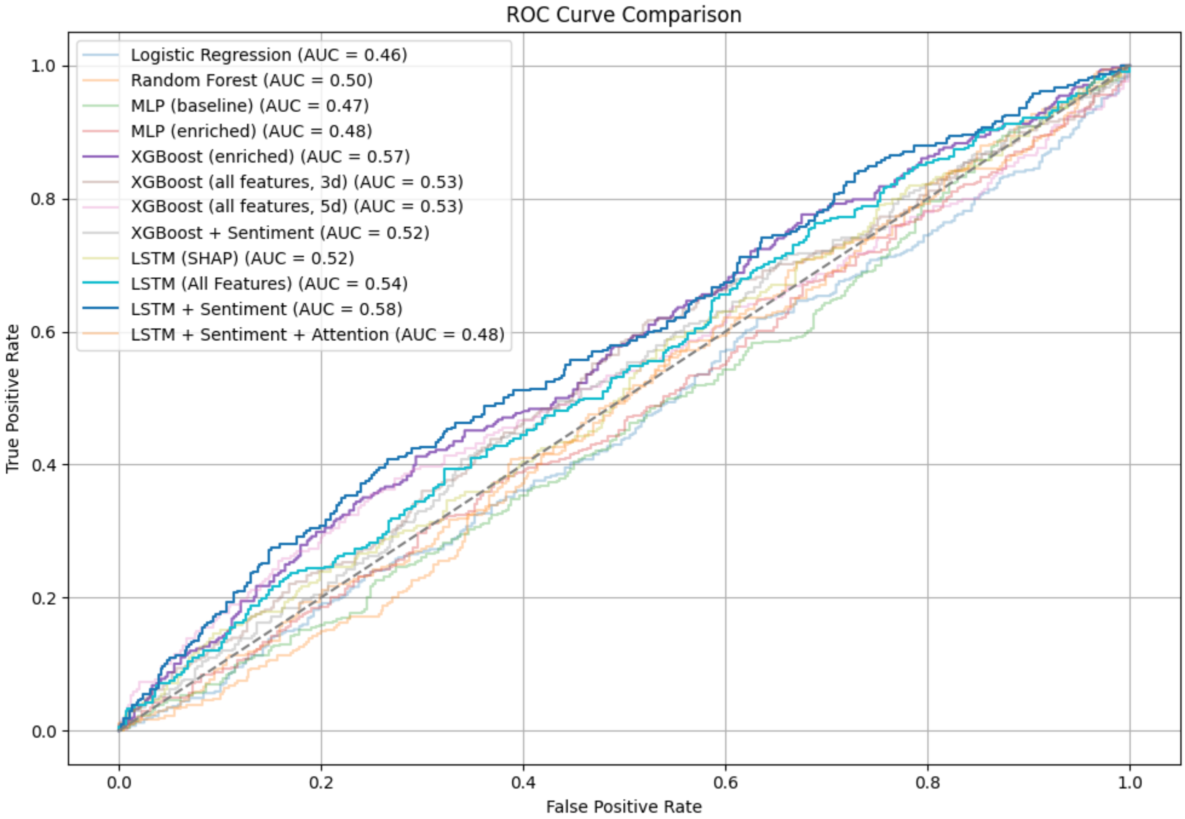

Figure 9: ROC curve comparison across all models.

The ROC plot confirms that XGBoost (10 features) and LSTM with sentiment have some of the highest AUC values, aligning with the table. Most other models cluster near the diagonal, suggesting limited ability to distinguish between upward and downward price movement.

8 Discussion and Limitations

In this project, we tested different models and feature sets to predict the short-term direction of Apple's stock. We started with simple models like Logistic Regression and Random Forest, and moved to more complex ones like LSTM with sentiment and attention.

One of the key observations was that no model was clearly the best across all metrics. Traditional models like Random Forest did surprisingly well in terms of balance (F1 and AUC), while deep learning models gave mixed results. For example, LSTM with sentiment had one of the highest AUC scores, but its F1 score was lower than expected.

We also saw that **feature selection** had a big impact. The six features chosen based on SHAP values worked quite well in several models. Adding a few more features (total of 10) helped slightly in some cases. When we included sentiment features, the results were mixed — in some cases, they improved the prediction, but not always.

Limitations

- The target variable is simplified — it only checks if the price goes up after 3 or 5 days. Real market behavior is more complex.
- All models were trained on past data. There's always the risk that market conditions change and models won't work well in the future.
- Sentiment features were basic — daily averages of polarity. We didn't capture the full meaning or importance of news headlines.
- We only looked at Apple. Results may be different for other companies or sectors.

Future Work

In the future, we could try:

- Using more detailed sentiment features (e.g., from large language models)
- Predicting different types of outcomes (e.g., exact return values or multi-class direction)
- Testing our models on other stocks or building a model that works across many companies
- Adding more data sources — like economic indicators or social media

To sum up, this project gave us a full pipeline — from feature engineering to modeling and evaluation — and showed how different choices in features and models affect stock prediction performance.

References

[1] Lundberg, S.M. and Lee, S.I., 2017. A Unified Approach to Interpreting Model Predictions. *Advances in Neural Information Processing Systems*, 30.

[2] Yfinance Library: `https://github.com/ranaroussi/yfinance`

[3] Apple Historical Financial News: `https://www.kaggle.com/datasets`