Predicting Stock Price Movement with Machine Learning

Danny Suradja
Master of Data Science
University of California, Irvine
dsuradja@uci.edu

Michael Cao
Master of Data Science
University of California, Irvine
qmcao@uci.edu

Richard Wicaksono
Master of Data Science
University of California, Irvine
wicaksor@uci.edu

***Abstract*—This project investigates the prediction of Costco (COST) stock price movements using machine learning techniques. We applied classical machine learning models (Random Forest and XGBoost) and advanced neural network methods (TensorFlow and PyTorch), incorporating technical indicators derived from historical stock data as well as sentiment analysis based on financial news articles. Our experiments demonstrate that neural network approaches significantly outperform classical machine learning models in terms of predictive accuracy. Interestingly, incorporating news sentiment features provided minimal performance improvement, suggesting the need for further refinement in sentiment feature extraction and representation methods.**

***Index Terms*—Stock prediction, Random Forest, XGBoost, TensorFlow, PyTorch, machine learning, sentiment analysis**

I. Introduction

Predicting stock price movements is a critical task within financial analytics, offering significant implications for both traders and investors. Accurate prediction of stock price direction—whether prices will rise or fall—can yield substantial financial benefits. However, the complexity and inherent volatility of financial markets make stock prediction particularly challenging. Numerous factors such as economic indicators, market trends, investor sentiment, corporate earnings reports, geopolitical events, and news headlines profoundly influence stock prices.

This study specifically targets predicting Costco (COST) stock price movements, employing multiple machine learning models to predict whether the stock's closing price will increase or decrease relative to the previous trading day. Costco was selected due to its substantial market presence, strong historical performance, and the availability of extensive historical data. Furthermore, Costco serves as an ideal candidate for demonstrating the robustness and predictive potential of advanced machine learning methods in real-world financial contexts.

II. Related Works

Extensive prior research explores using various machine learning models to predict stock prices. Classical approaches such as Random Forest and XGBoost have shown success due to their simplicity and interpretability. Recent trends also demonstrate that neural networks, including TensorFlow and PyTorch implementations, capture complex patterns in stock market data effectively, often outperforming classical approaches.

III. Exploratory Data Analysis (EDA)

We began by analyzing various historical stock data (NVDA, TSLA, COST, META) retrieved from Yahoo Finance. The figures presented below illustrate exploratory data analysis conducted on a one-year period solely for example purposes and clarity of visualization. However, for actual modeling purposes, historical data spanning a ten-year period was utilized. The following figures Key features include:

Closing Price: The closing price is the last price at which the stock is traded during the regular trading day. A stock's closing price is the standard benchmark used by investors to track its performance over time (Fig. 1).

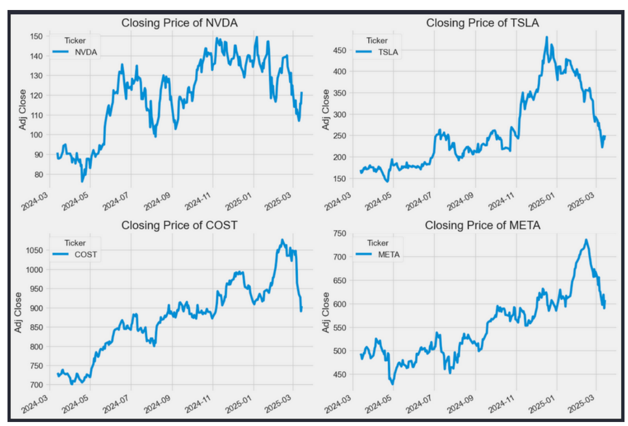

Fig. 1. Daily closing prices for selected stocks.

Volume: Volume is the amount of an asset or security that changes hands over some period of time, often over the course of a day. For instance, the stock trading volume would refer to the number of shares of security traded between its daily open and close. Trading volume, and changes to volume over the course of time, are important inputs for technical traders. (Fig. 2).

Daily Returns: Daily Returns: Daily returns represent the percentage change in a stock's closing price from one trading day to the next. Mathematically, it is calculated as the difference between today's closing price and the previous

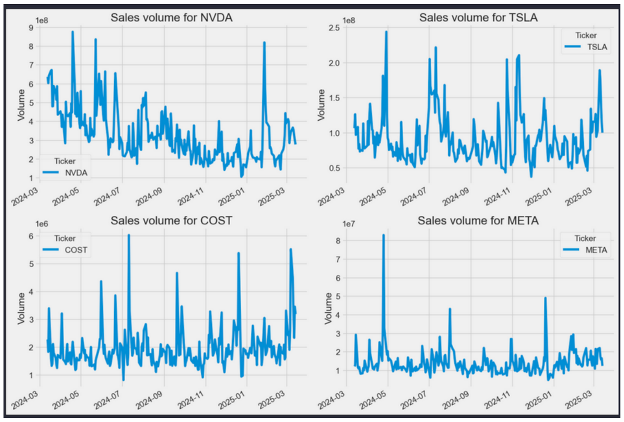

Fig. 2. Daily sales volume for selected stocks (in 100 millions).

day's closing price, divided by the previous day's closing price, expressed as a percentage. Daily returns are a critical measure for understanding stock volatility, as they quantify how rapidly and significantly stock prices fluctuate over short periods. By analyzing the distribution and frequency of daily returns (Fig.3, Fig.4), investors can assess risk levels, identify volatility patterns, and make informed decisions regarding investment strategies. High volatility, characterized by substantial fluctuations in daily returns, typically indicates greater market uncertainty, while low volatility suggests a more stable and predictable stock performance.

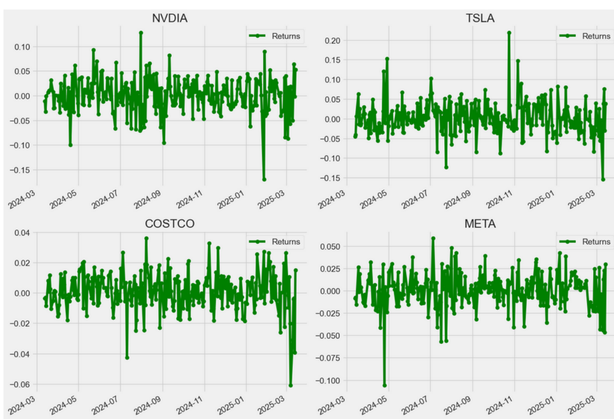

Fig. 3. Daily returns (percentage) for selected stocks.

Technical Indicators: Technical indicators, including Simple Moving Average (SMA), were utilized. The moving average (MA) is a simple technical analysis tool that smooths out price data by creating a constantly updated average price. The average is taken over a specific period of time, like 10 days, 20 minutes, 30 weeks, or any time period the trader chooses. (Fig. 5).

IV. Methods

We collected historical stock data for Costco (COST) from Yahoo Finance (2014-2024), then engineered features including moving averages (SMA), RSI, MACD, volatility,

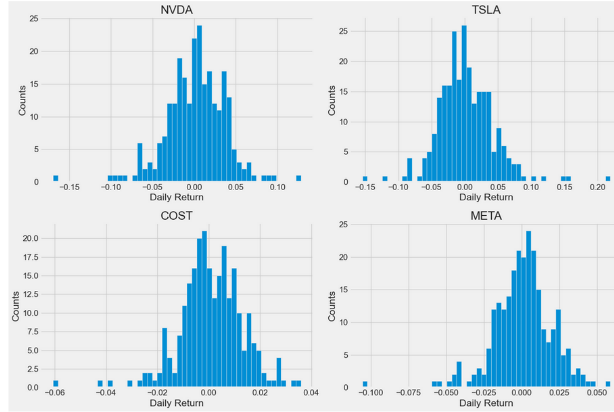

Fig. 4. Daily average return counts for selected stocks.

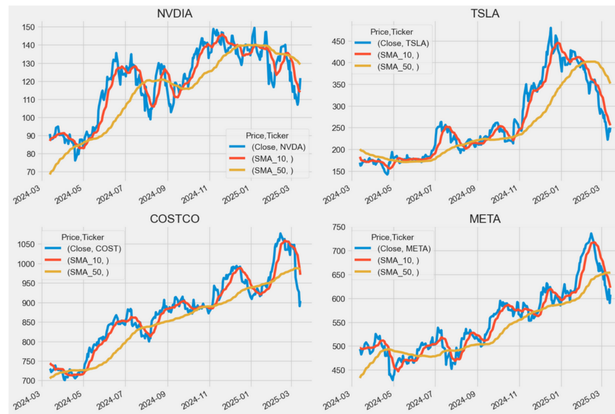

Fig. 5. Technical indicators (Close, SMA_10, SMA_50).

and returns. Two feature sets were evaluated: one with only technical indicators, and another incorporating news sentiment. Four modeling pipelines were developed:

- Classical ML: Random Forest, XGBoost (with hyperparameter tuning using TimeSeriesSplit cross-validation)
- Neural Networks: TensorFlow and PyTorch-based deep learning models

Models were evaluated on accuracy, precision, recall, and F1-score.

A. *XGBoost and Random Forest Methods*

In our study, we decided to use Random Forest and XGBoost as our base models for stock prediction. Random Forest, an ensemble method that builds multiple decision trees and averages their predictions, is well known for its ability to handle complex, non-linear relationships in the data. Random Forest provides measures of feature importance, which can help us understand which technical indicators or sentiment measures most influence the predictions.

XGBoost, on the other hand, is a boosting algorithm that builds models sequentially. Each new tree is constructed to correct the errors of the previous ones, which allows the model to capture subtle patterns in the data. XGBoost also incorporates regularization techniques that help prevent overfitting. Its speed and scalability make it suitable in financial model, and it has shown strong performance in predictive tasks.

We want to compare the accuracy and F1 score of the two models to maximize profit as a trader. After training the model, we achieved 60% accuracy of correctly predicting whether the closing price of COST stock is higher than the previous day. Adding news sentiment, we found that xG boost is performing the best, with the highest accuracy and F1 score of these 2 methods. However, the accuracy is only about 67%.

B. Neural Network Approach

We built two neural network pipelines—one using TensorFlow and another using PyTorch—to predict stock movements. In both cases, the underlying idea was to model the non-linear relationships inherent in the data, which includes technical indicators and sentiment scores. The data is first standardized and split into training and test sets while preserving the temporal order to avoid look-ahead bias.

For the TensorFlow pipeline, we designed a simple feed-forward neural network with two hidden layers. The model uses ReLU activations in the hidden layers and a sigmoid activation in the output layer to predict the binary target. The training process, driven by the Adam optimizer and binary cross entropy loss, is monitored over multiple epochs using a validation split to ensure the model generalizes well.

In the PyTorch pipeline, we adopted a similar network architecture but implemented the training loop manually. Data is converted to tensors and batched with DataLoaders. The network is trained by iterating over mini-batches, calculating loss, and updating weights via the Adam optimizer. This approach offers more fine-grained control over the training process, allowing us to track performance metrics such as loss and accuracy on both the training and validation sets throughout the epochs.

By using these complementary deep learning approaches, we aim to capture complex patterns that may be overlooked by traditional models. Each pipeline was tuned to ensure that the networks learn the underlying data structure effectively, thus providing robust predictions on stock movements.

C. finBERT Integration

Additionally, in order to enhance the robustness of our stock prediction model, we incorporated a sentiment score as an additional feature alongside stock movement data. Ideally, we would retrieve a real-time news dataset through APIs such as NewsAPI or GNews; however, due to the limitations imposed by free access, we instead employed an open-source news dataset to derive sentiment scores for articles related to our stock of interest. To obtain these sentiment scores, we utilized finBERT— a natural language processing model specifically trained to analyze the sentiment of financial texts. FinBERT assigns a sentiment score ranging from -1 to 1, where -1 indicates extreme negative sentiment, +1 indicates extreme positive sentiment, and 0 represents neutrality. In our model, we feed a rolling mean of sentiment score over 5 days. This tailored sentiment analysis is particularly suitable for our application, as it captures the tone of financial news that may influence market movements.

V. Experimental Results

Model performances are summarized in Table I.

TABLE I
PERFORMANCE COMPARISON OF MODELS

Model	Condition	Acc.	Prec.	Recall	F1-score
Random Forest	Without News Sentiment	0.66	0.67	0.66	0.65
	With News Sentiment	0.63	0.63	0.63	0.63
XGBoost	Without News Sentiment	0.66	0.68	0.66	0.65
	With News Sentiment	0.67	0.68	0.67	0.67
TensorFlow	Technical Indicators	>0.98	–	–	–
PyTorch	Technical Indicators	>0.98	–	–	–

Our results revealed a clear advantage of neural networks over classical methods, with TensorFlow and PyTorch achieving significantly higher accuracy, suggesting that complex relationships in market data are better captured by deep learning models (Fig.6, Fig.7). Surprisingly, incorporating news sentiment scores provided only marginal improvement for classical models, indicating that sentiment feature representation may require further improvement or the sentiment itself had minimal predictive power for this particular stock.

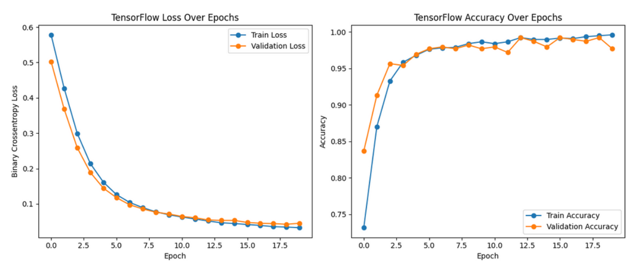

Fig. 6. TensorFlow accuracy and loss over epochs.

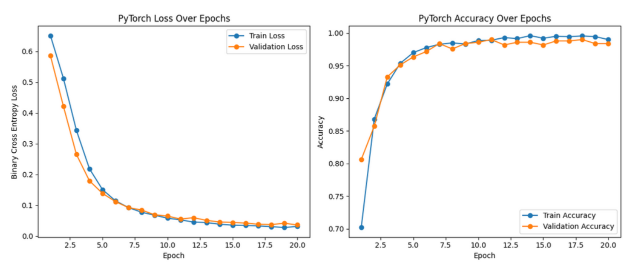

Fig. 7. PyTorch accuracy and loss over epochs.

VI. Conclusion

In conclusion, this study underscores the efficacy of advanced neural network models (TensorFlow and PyTorch) in accurately predicting Costco stock price movements when compared to classical methods like Random Forest and XGBoost. Although sentiment analysis features provided minimal incremental predictive value, the observed slight performance improvement indicates untapped potential, pending further methodological improvements.

Future research should explore more advanced NLP techniques for sentiment extraction, evaluate a broader array of external market indicators, experiment with ensemble methods combining classical and neural network predictions, and investigate the implementation of real-time adaptive prediction systems that dynamically respond to rapidly changing market conditions.

TEAM CONTRIBUTIONS

This project was conducted collaboratively by:

- **ML Modeling**: Richard Wicaksono developed, optimized, and validated machine learning models, including classical (Random Forest, XGBoost) and neural network models (TensorFlow, PyTorch).
- **EDA and Technical Documentation**: Michael Cao conducted thorough exploratory data analysis, generated insightful visualizations, and documented the process comprehensively.
- **finBERT Engineering and Integration**: Danny Suradja managed sentiment analysis using finBERT, extracting and integrating sentiment scores into the ML models.

The team collaborated effectively through regular meetings to share insights, discuss model improvements, evaluate results collectively, and integrate all aspects into a cohesive and comprehensive final report.