

Boston University
Electrical & Computer Engineering

EC 464 Senior Design Project

AI Trading Platform

By: Team 6

Final Test Report

Adrian Pawlowski | ajp12@bu.edu
Gabriel Brown | gabrown@bu.edu
Yagiz Idlman | ykaan@bu.edu
Odilon Quevillon | odilon@bu.edu

1. Introduction

This report presents the final testing results of our AI Trading Platform, including three machine learning models (LSTM, XGBoost, and Reinforcement Learning), and evaluates their performance using a new dataset ranging from January 1, 2024 to April 1, 2025. It also includes live trading results collected over the past 7 trading days, along with an evaluation of the front-end and back-end integration.

The objective is to assess model performance across various financial metrics and confirm that they meet or exceed the measurable criteria outlined in our Final Lab Testing Plan.

2. Equipment and Setup

Required Materials

- **LSTM Model:**
 - lstm_model.py, train_lstm.py, backtest_lstm.py, lstm_backtest_results.ipynb
- **XGBoost Model:**
 - train_boost.py, backtest_boost.py, boost_backtest_results.ipynb
- **Reinforcement Learning Model:**
 - dqn_agent.py, train_rl.py, trading_env_shares.py, run_trained_agent.py, rl_backtest_results.ipynb
- **NLP Model:**
 - run_nlp.py
- **Live Trading Tools:**
 - create_csv.py, results.py, results_notebook.ipynb
- **Front-End/Back-End:**
 - React-based front-end files and Java back-end API server running via Gradle
- **Data:**
 - CSV files for AMZN, JNJ, JPM, KO, LMT, MSFT, NKE, NVDA, TSLA, XOM

Setup/Plan

Models were trained and backtested on **Google Colab** and a local machine. A separate local instance hosted the **backend API** and front-end client, with the **Predictions page** displaying daily model-generated signals.

3. Methodology

Testing Procedure

1. Train and backtest each model (LSTM, XGBoost, RL) across all 10 stocks.
2. Record and evaluate key metrics: realized profit/loss, percent gain, annualized return, and risk ratios.
3. Execute live trades over 7 trading days and capture actual performance.
4. Display extrapolated projections.
5. Run the API and front-end client locally.
6. Confirm live prediction delivery and front-end signal display.

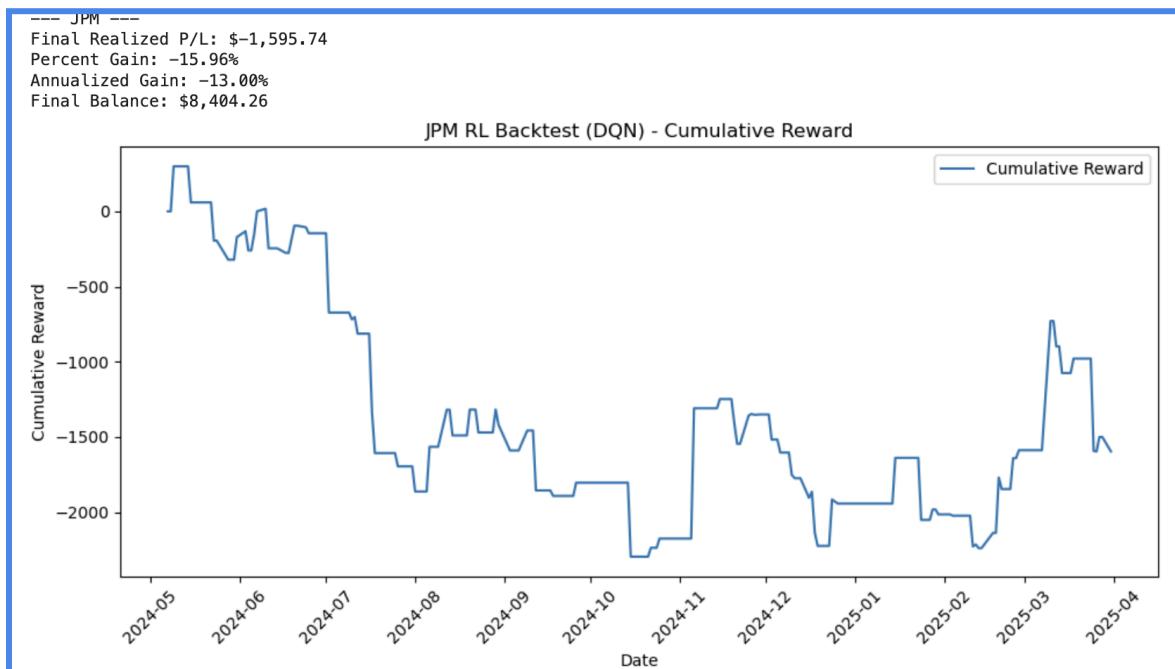
4. Measurable Criteria and Results

4.1 Reinforcement Learning Model Results

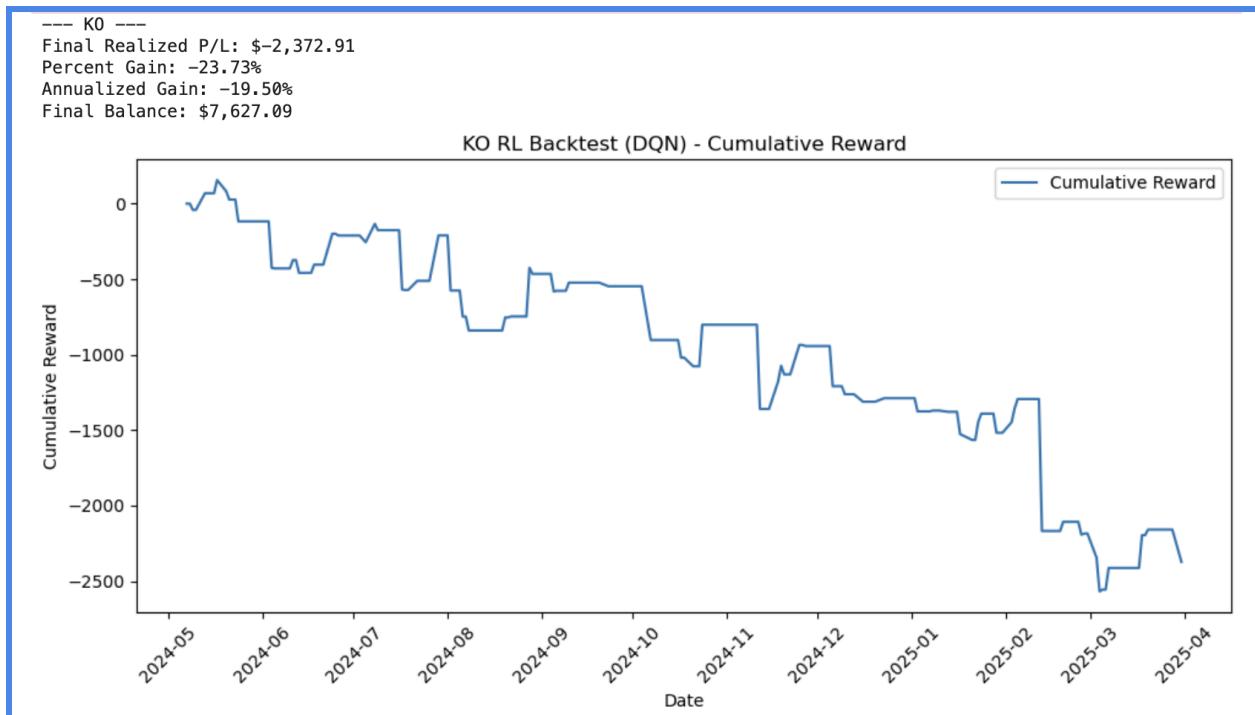
Metrics Evaluated:

- Final Realized P/L
- Percent Gain
- Annualized Gain
- Final Balance

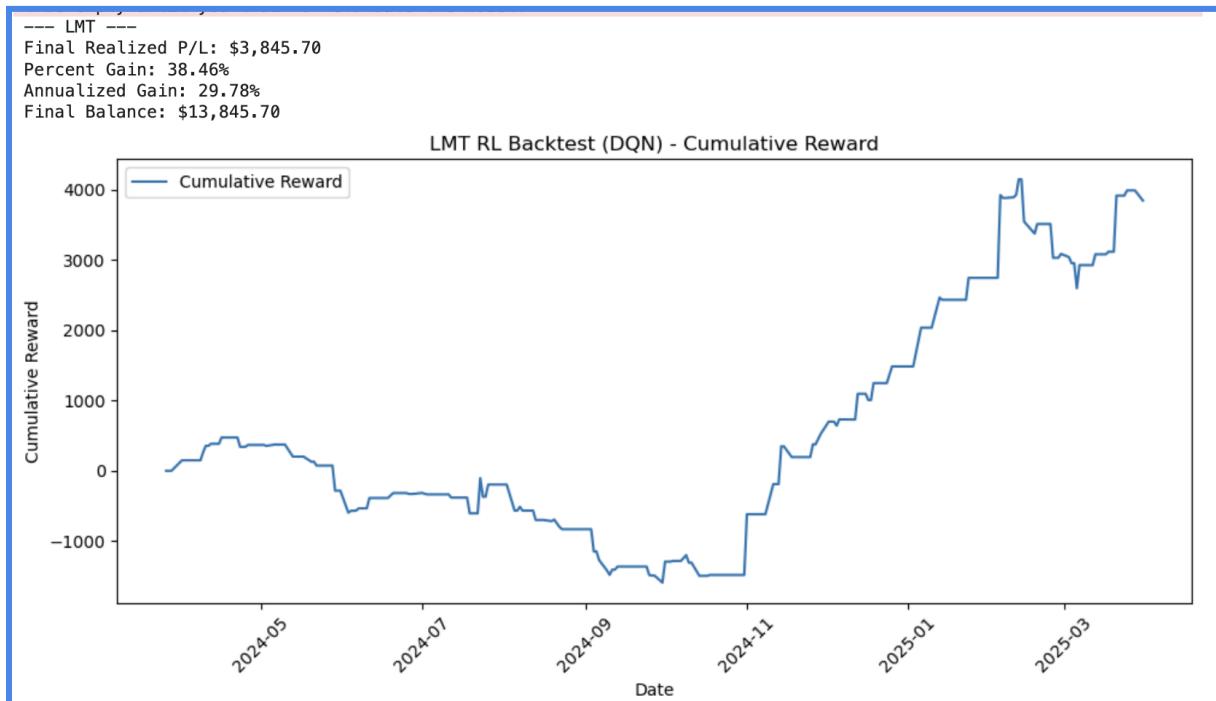
JP Morgan (JPM)



The Coca-Cola Company (KO)



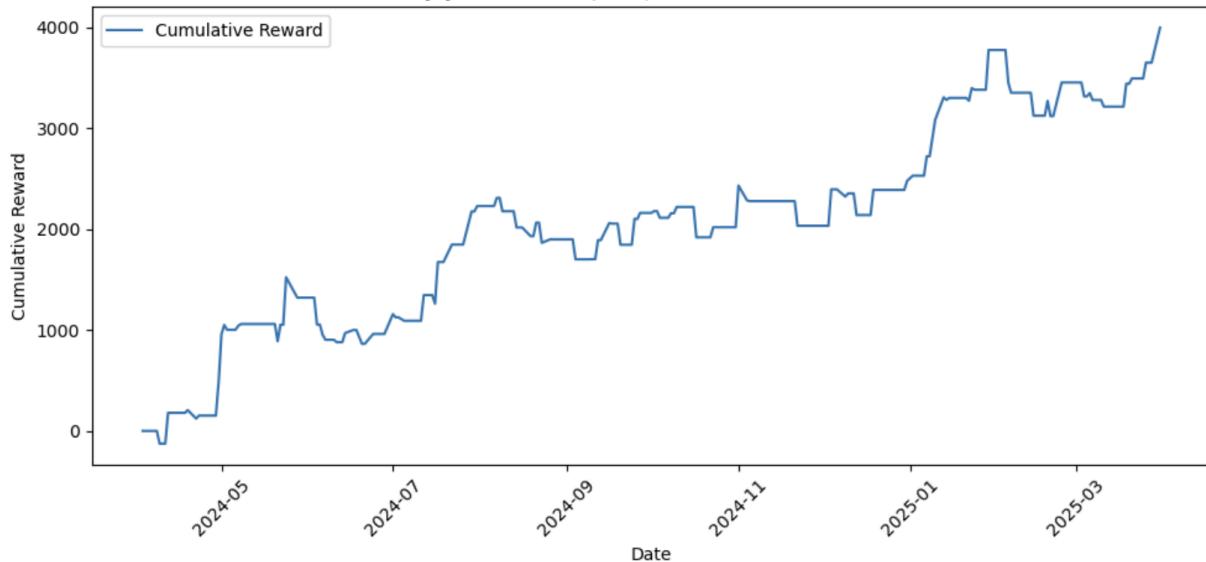
Lockheed Martin Corp (LMT)



Johnson & Johnson (JNJ)

--- JNJ ---
Final Realized P/L: \$3,997.35
Percent Gain: 39.97%
Annualized Gain: 30.91%
Final Balance: \$13,997.35

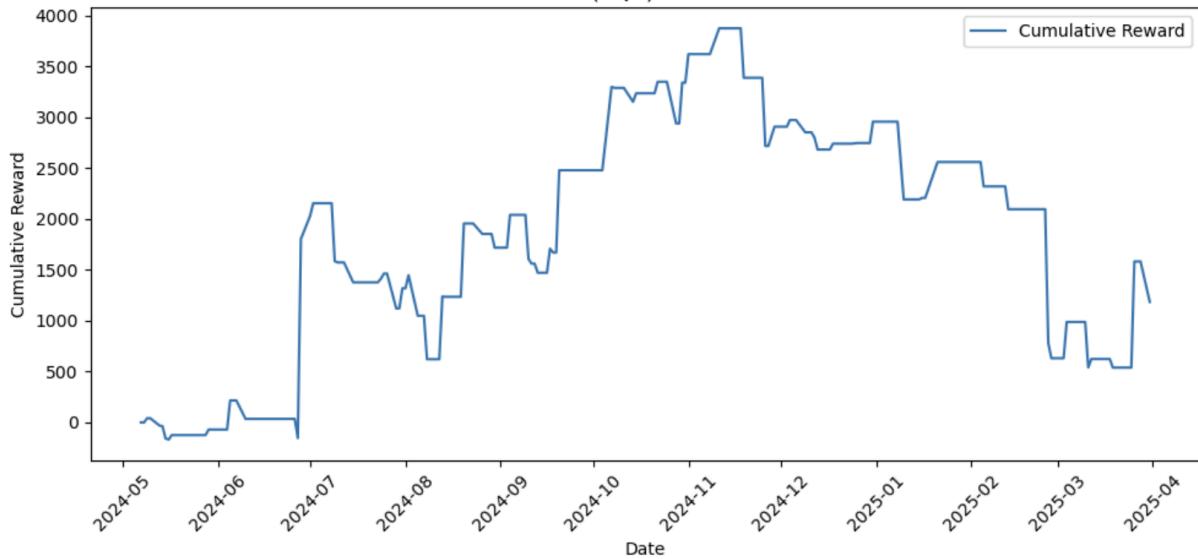
JNJ RL Backtest (DQN) - Cumulative Reward



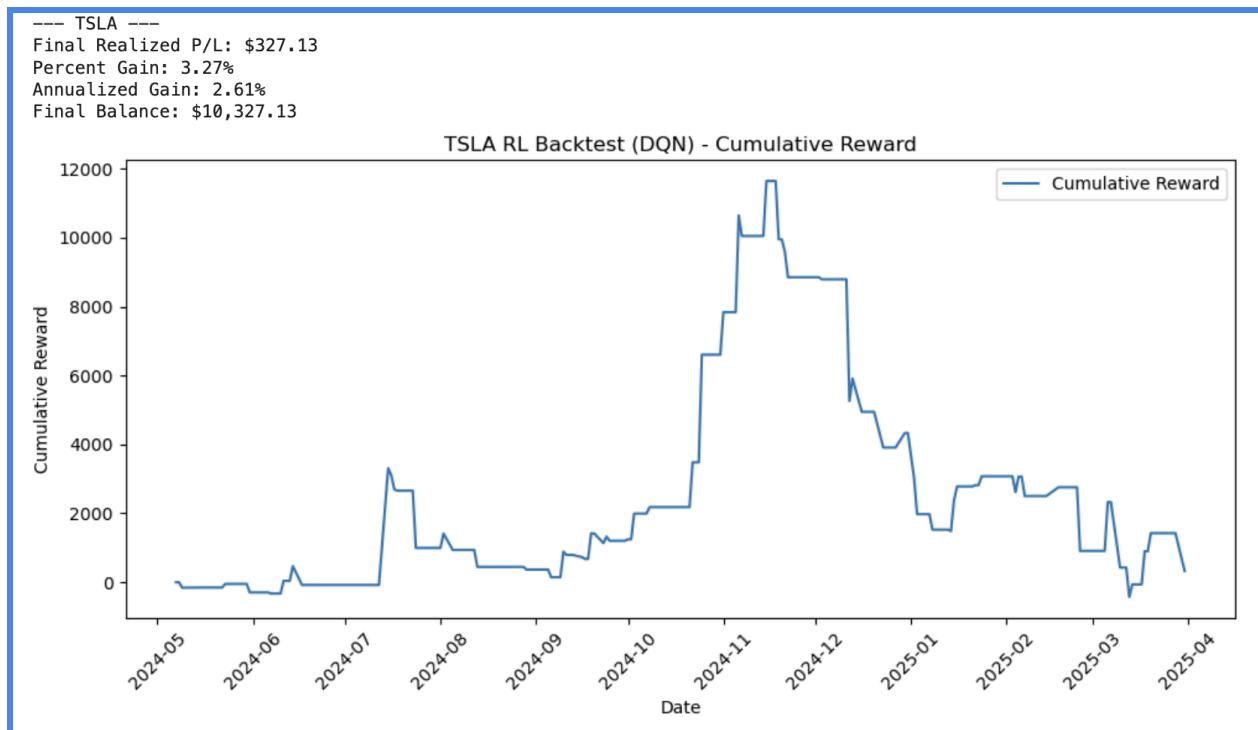
Nike Inc (NKE)

--- NKE ---
Final Realized P/L: \$1,185.26
Percent Gain: 11.85%
Annualized Gain: 9.39%
Final Balance: \$11,185.26

NKE RL Backtest (DQN) - Cumulative Reward



Tesla (TSLA)



4.2 Reinforcement Learning Aggregate Results

===== PORTFOLIO AGGREGATE =====

Number of Tickers: 6

Initial Total Portfolio: \$60,000.00

Final Total Portfolio: \$65,386.79

Total Profit: \$5,386.79

Percent Gain: 8.98%

Annualized Gain: 7.13%

4.3 XGBoost Model Results

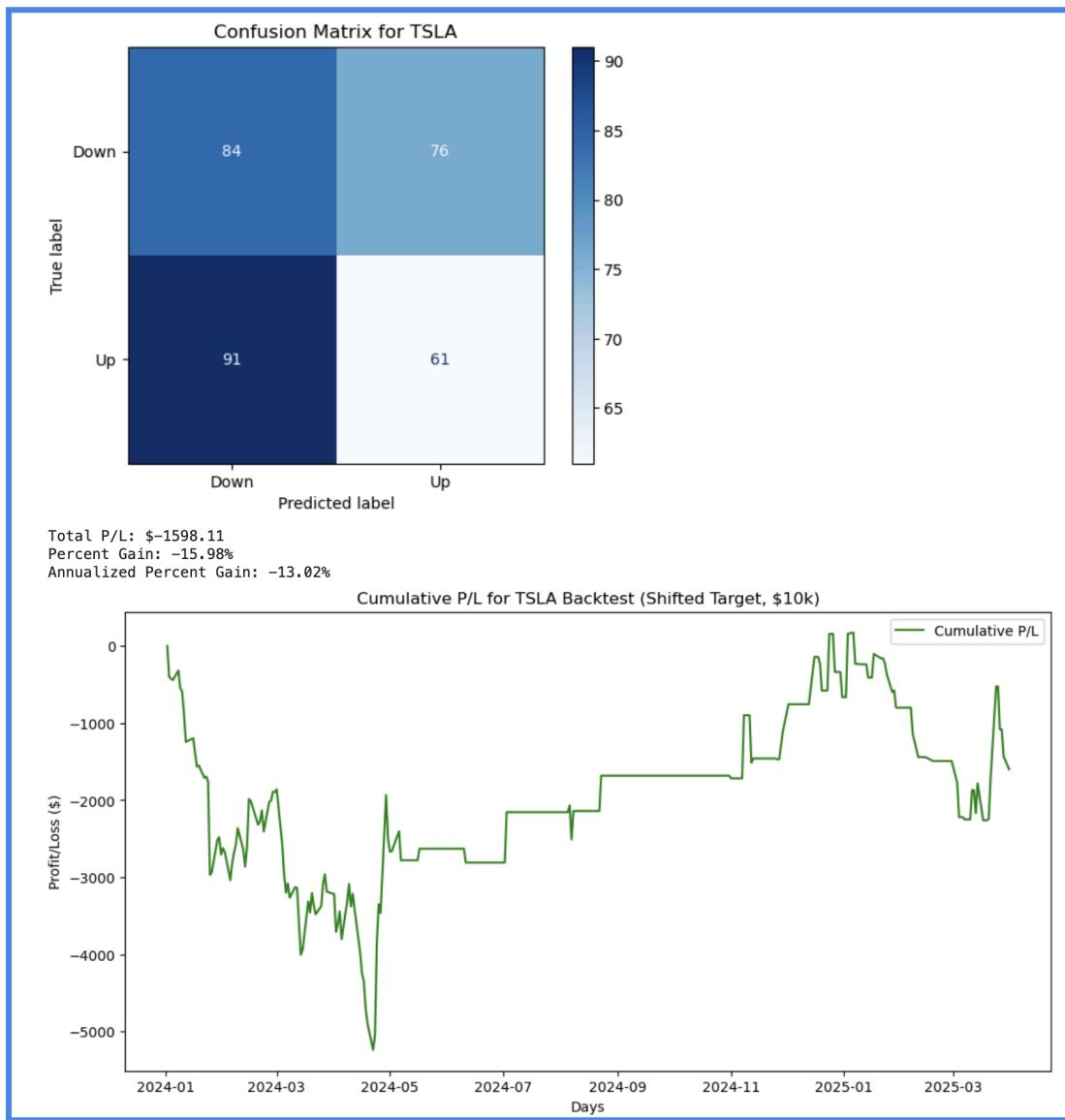
Individual Model Criteria:

- Total P/L
- Percent Gain
- Annualized Gain

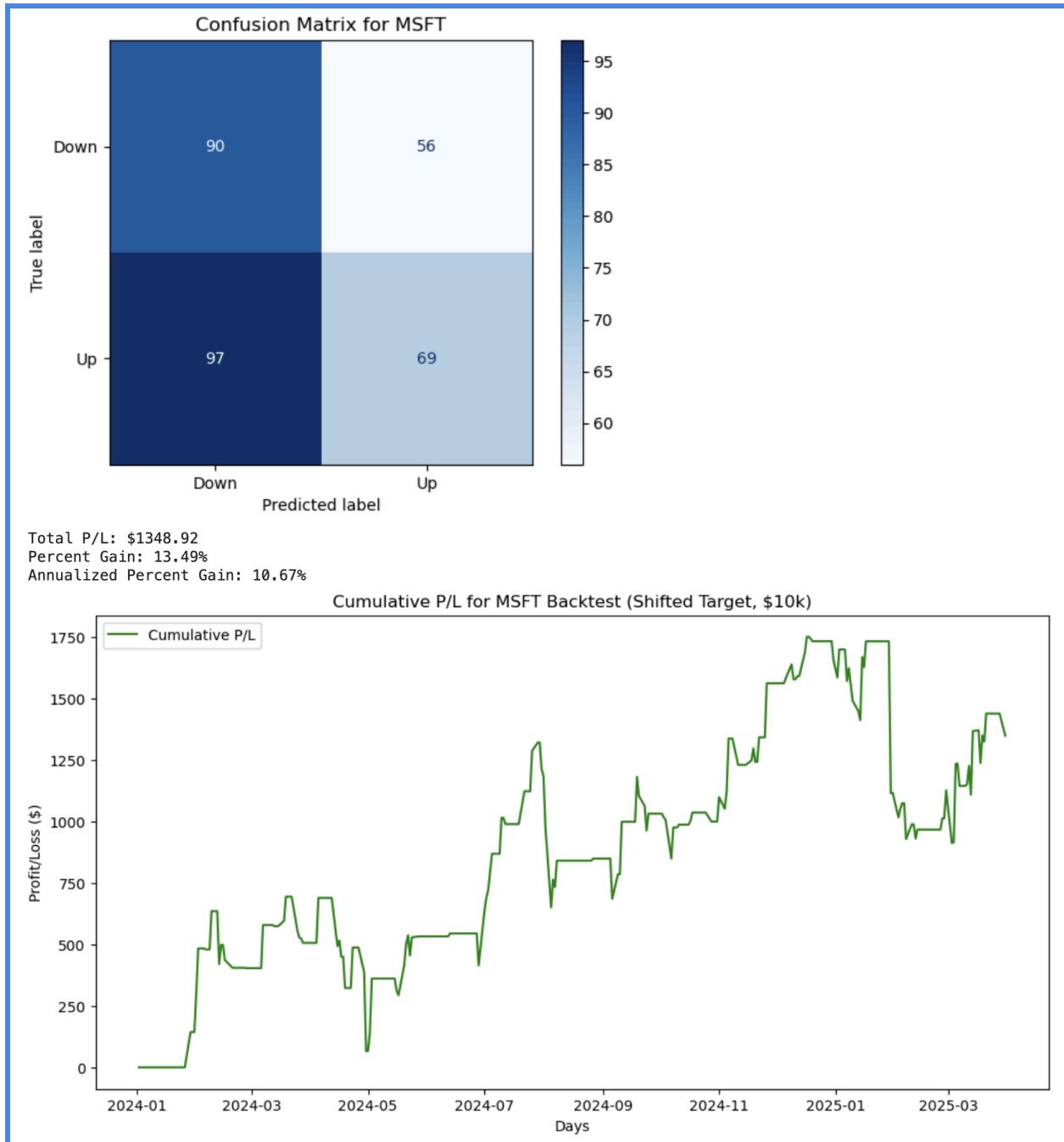
Aggregate Criteria:

- Total P/L
- Percent Gain
- Annualized Gain

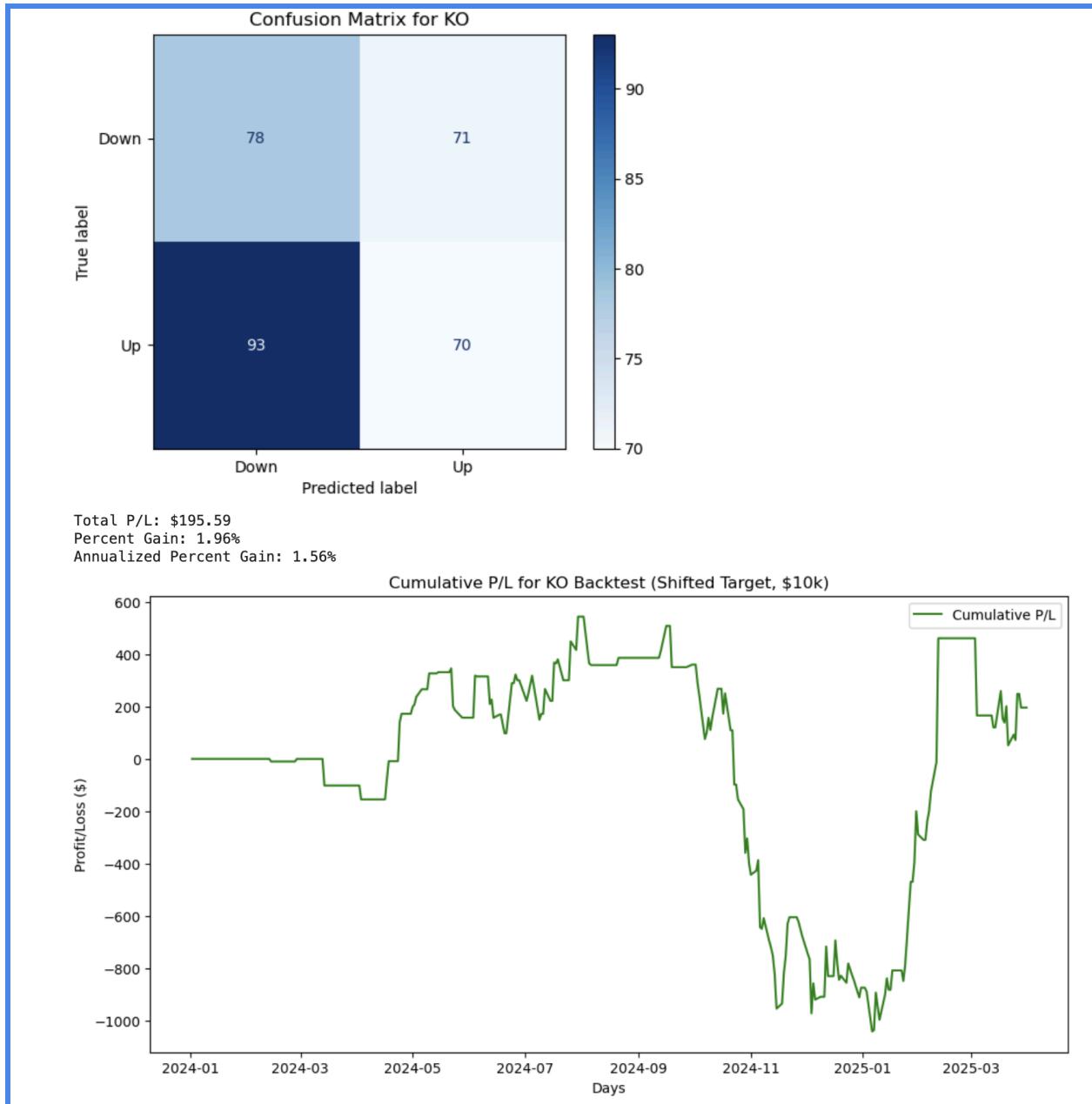
Tesla (TSLA)



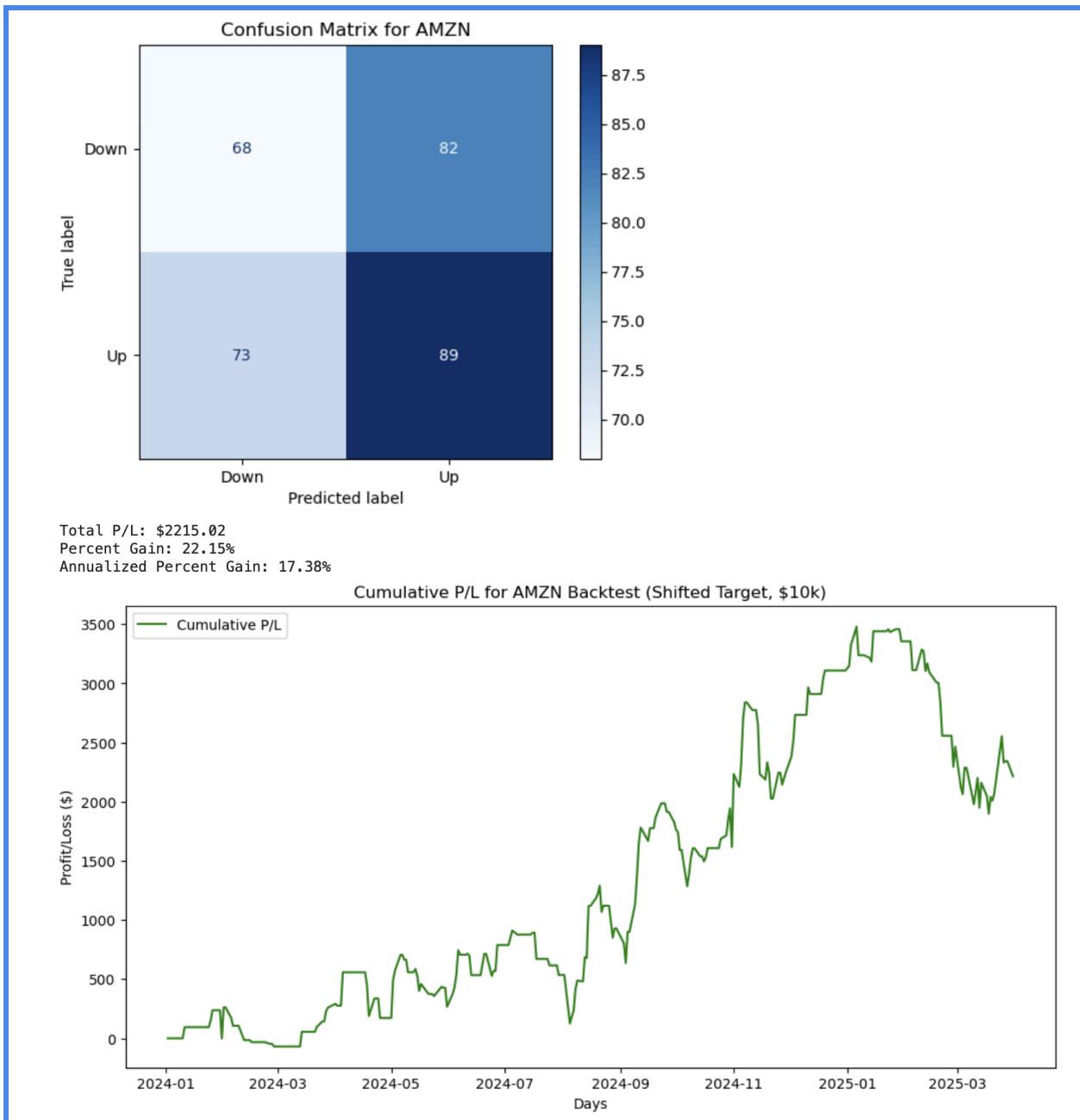
Microsoft (MSFT)



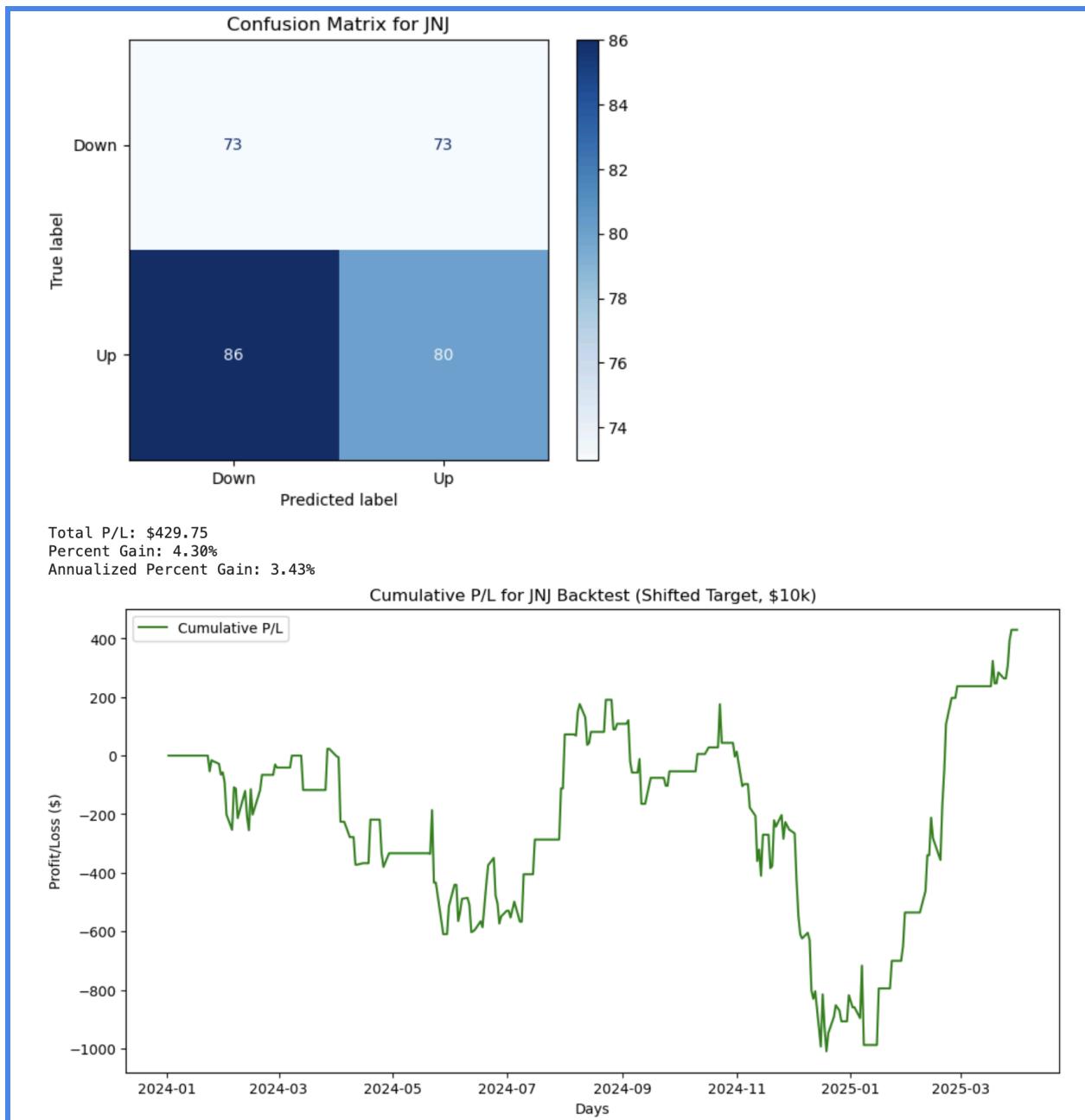
The Coca-Cola Company (KO)



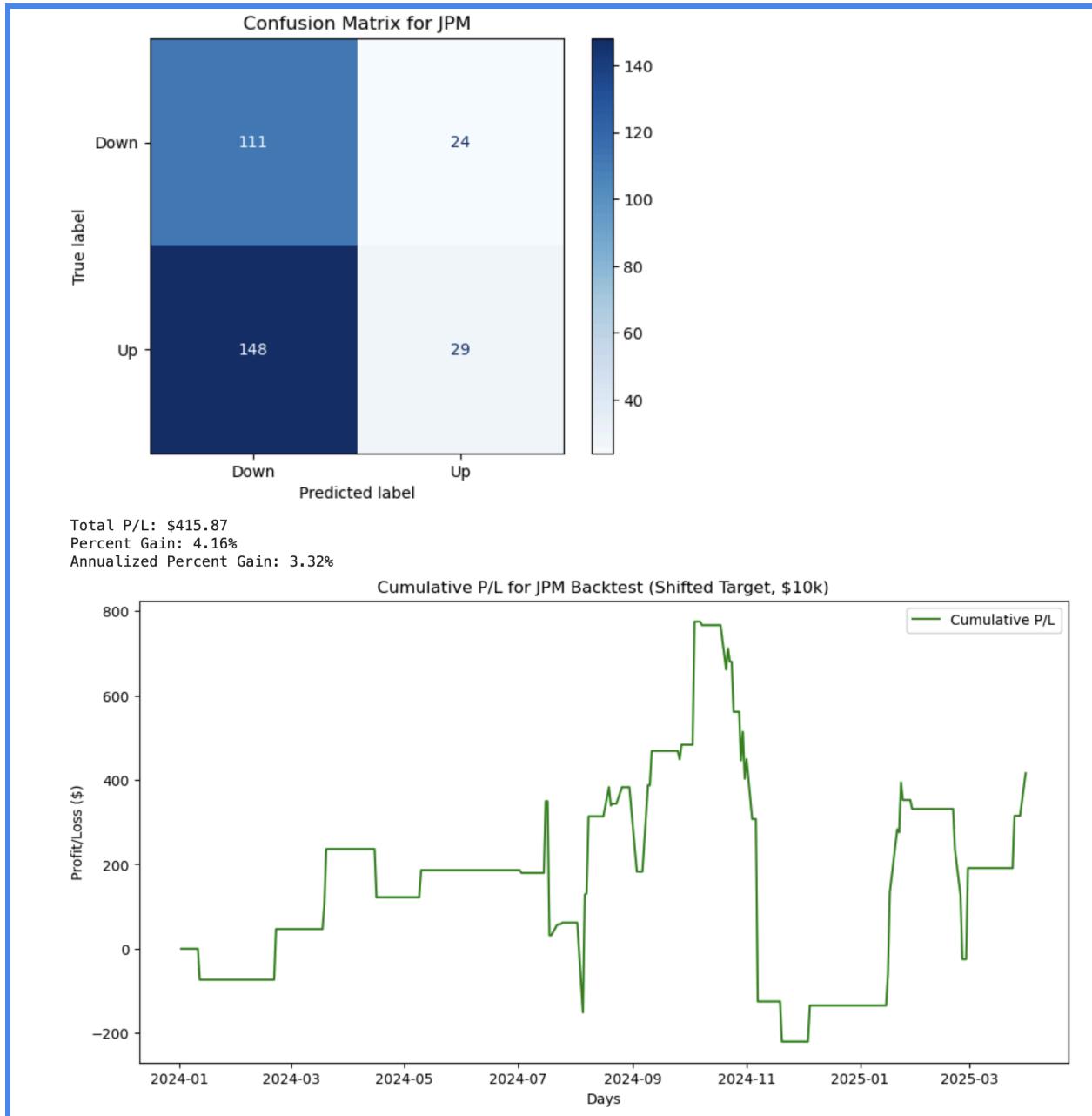
Amazon (AMZN)



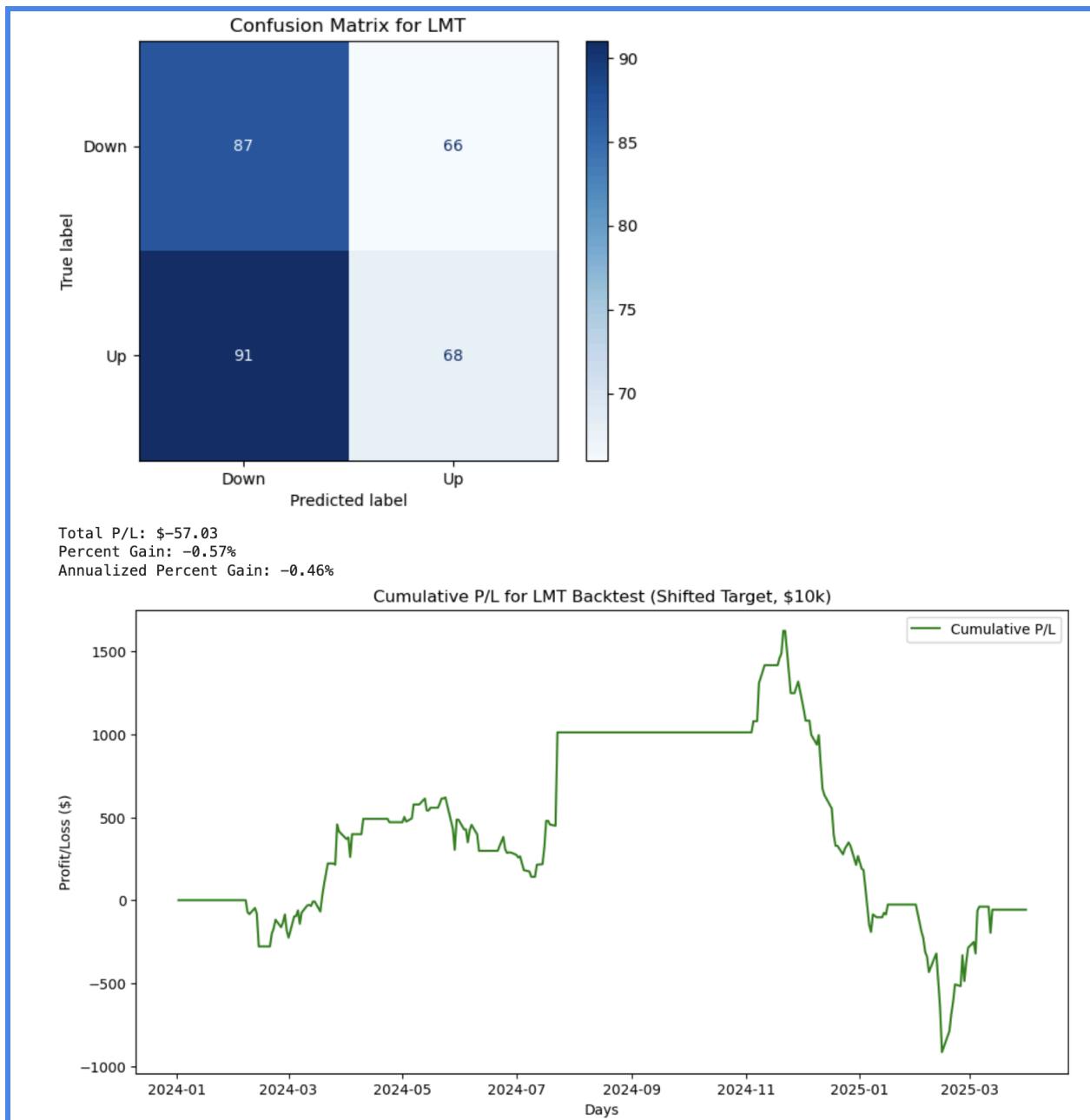
Johnson & Johnson (JNJ)



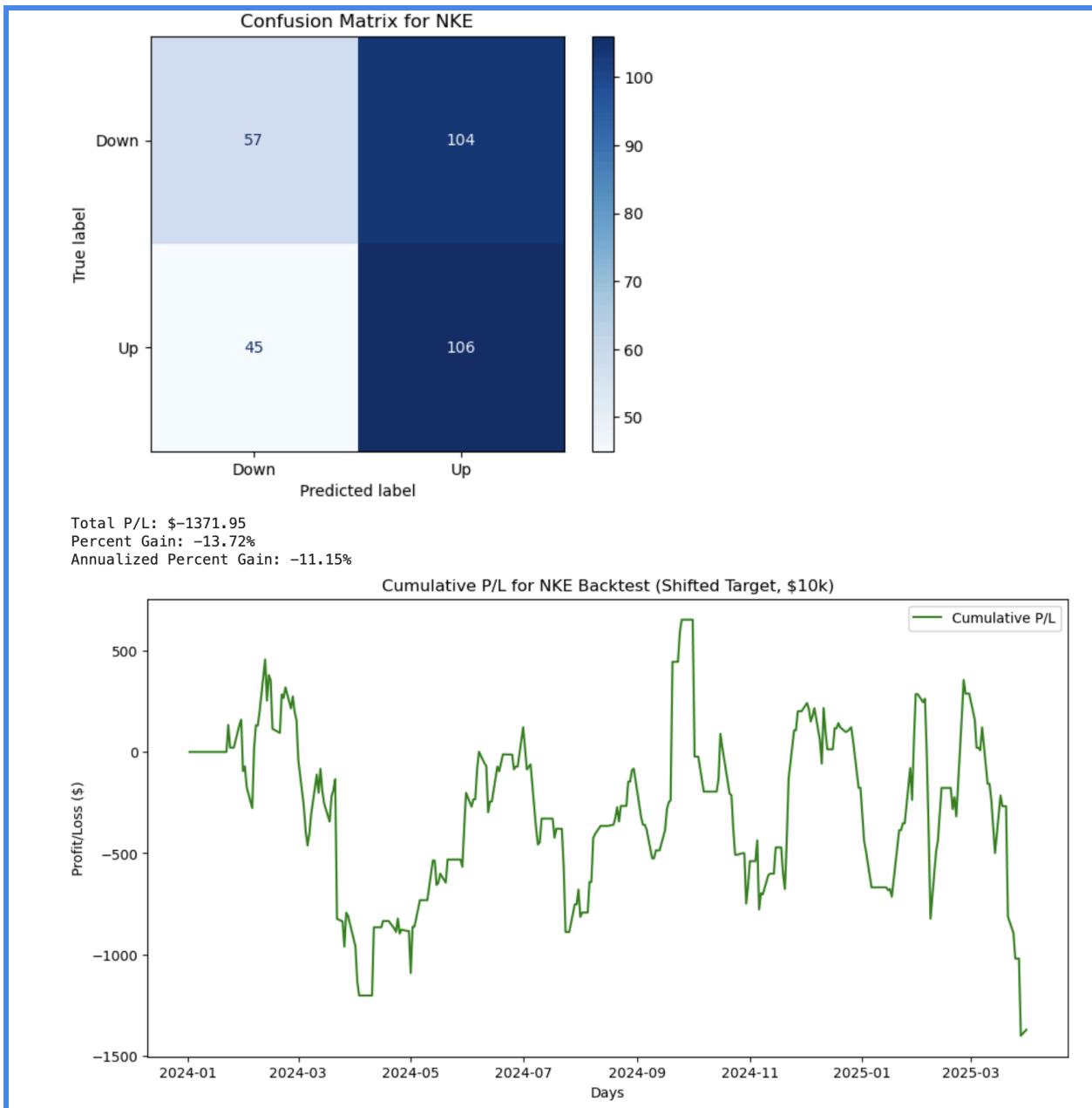
JP Morgan (JPM)



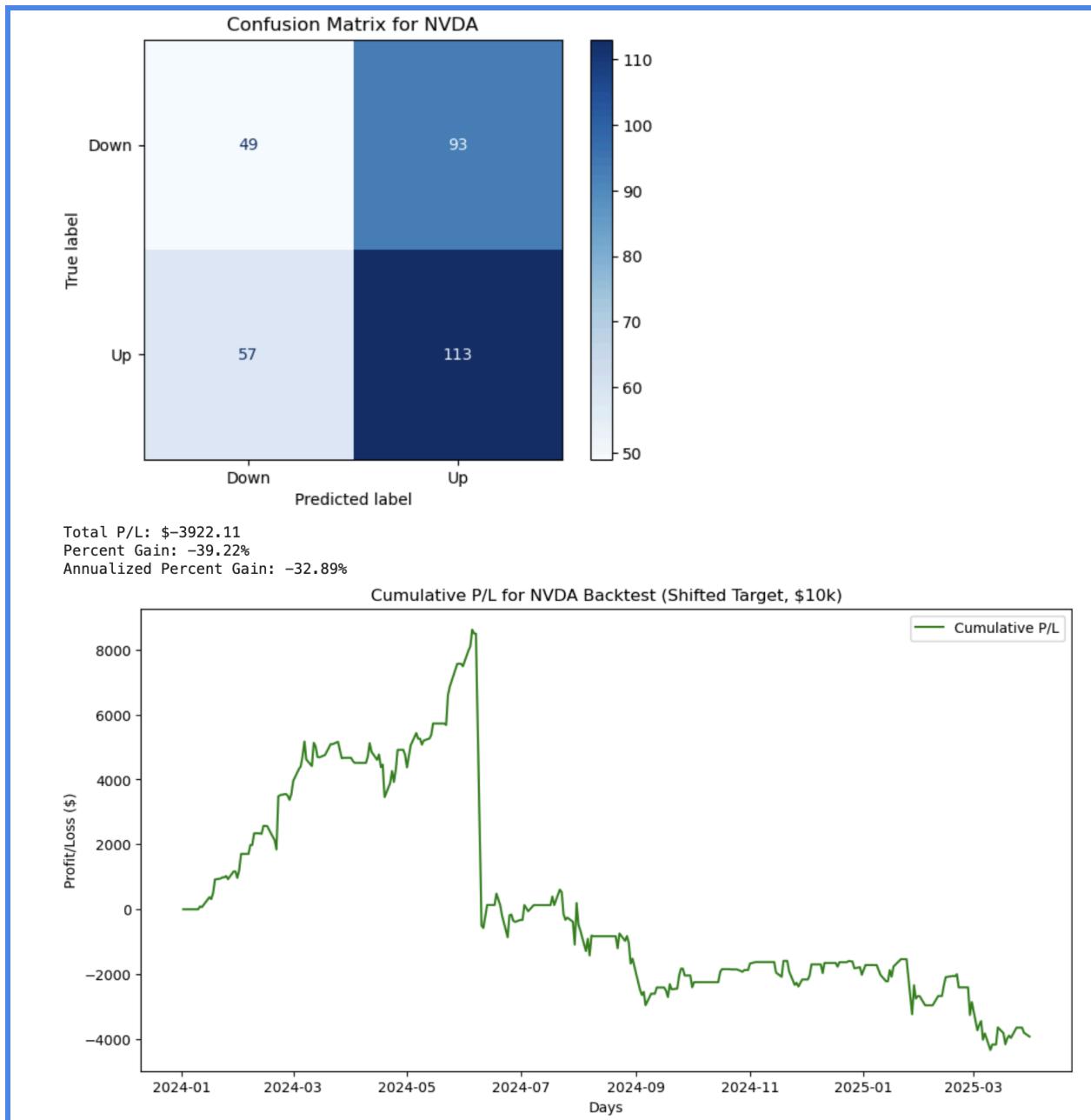
Lockheed Martin Corp (LMT)



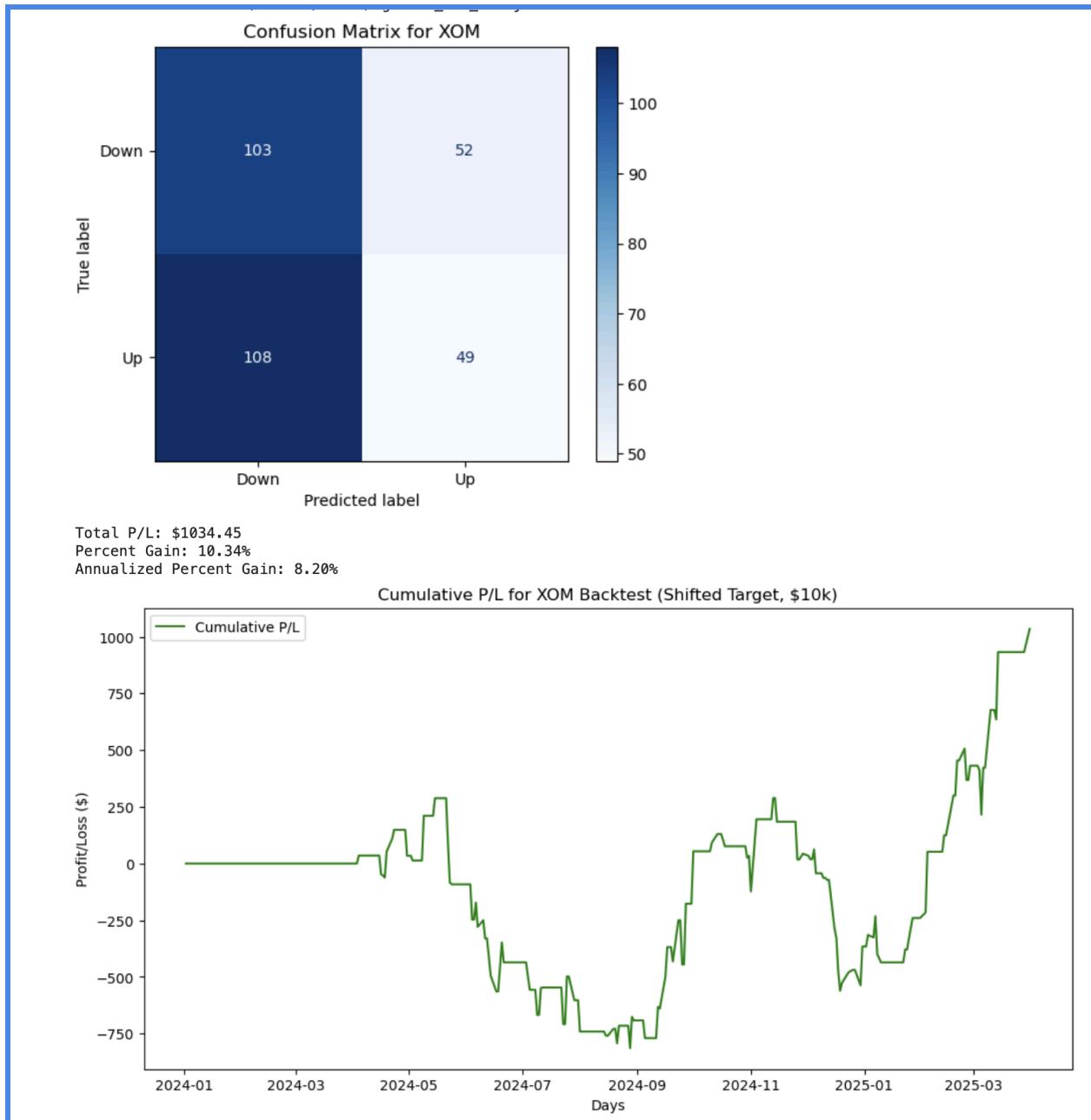
Nike Inc (NKE)



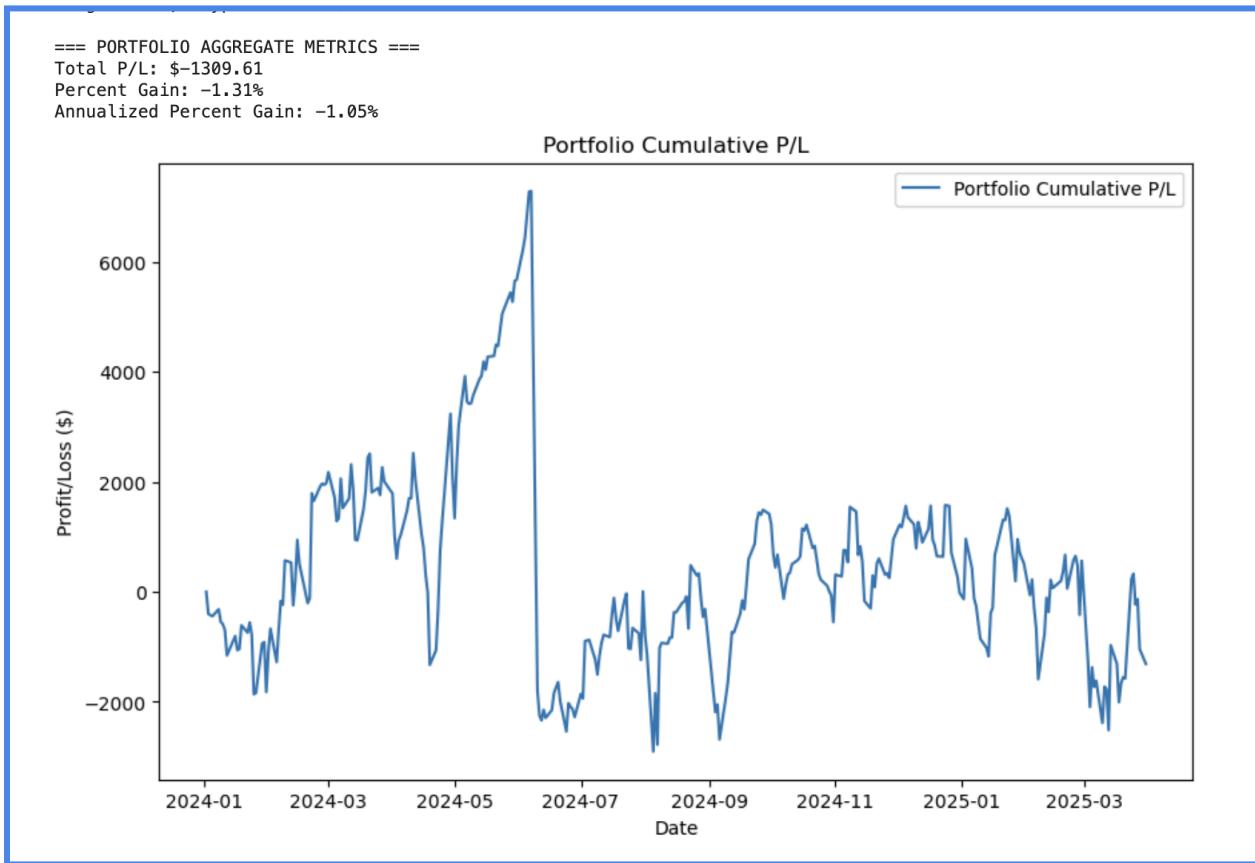
NVIDIA (NVDA)



Exxon Mobil (XOM)



4.4 XGBoost Aggregate Results



4.5 LSTM Model Results

Metrics Evaluated:

- MAE, RMSE, MAPE, R²
- Total P/L
- Overall Gain
- Annual Gain
- Sharpe Ratio

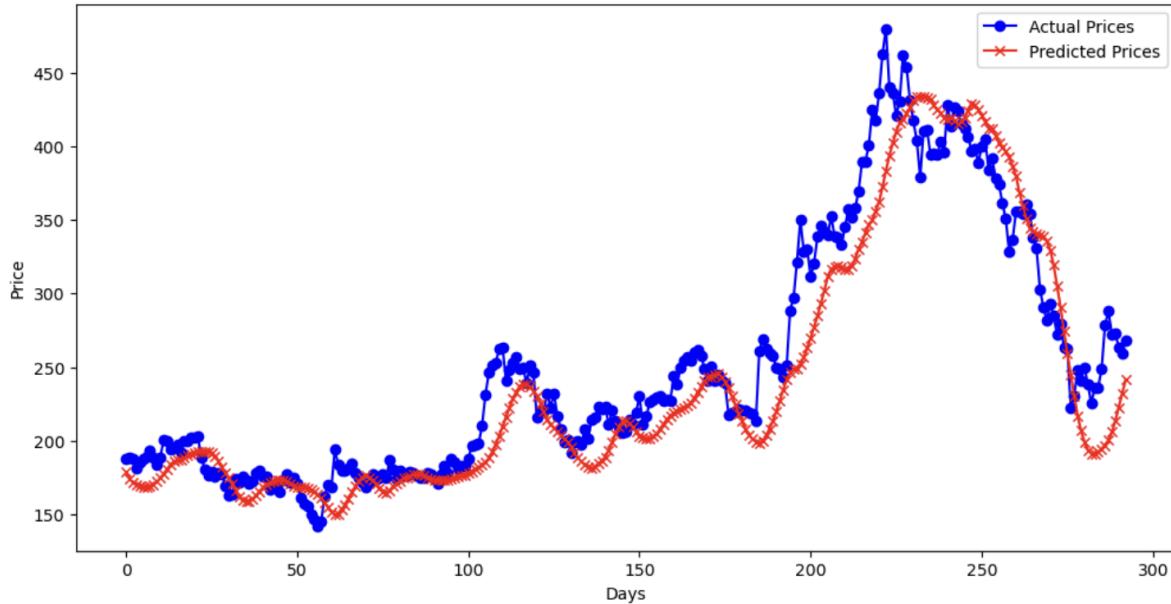
Portfolio Aggregate:

- Total P/L
- Overall Gain
- Annual Gain
- Sharpe Ratio

Tesla (TSLA)

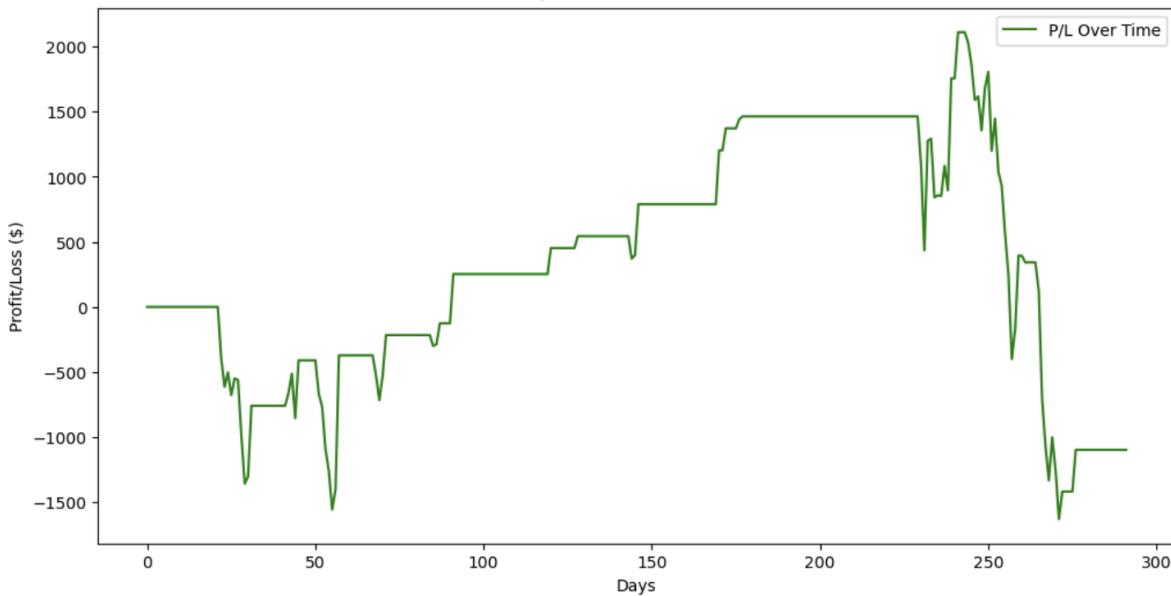
```
--> Backtesting Results for TSLA (2024-01-01 to 2025-04-01):
MAE: 22.19, RMSE: 29.54, MAPE: 8.38%, R2: 0.8725
```

Backtesting: Actual vs Predicted Prices for TSLA

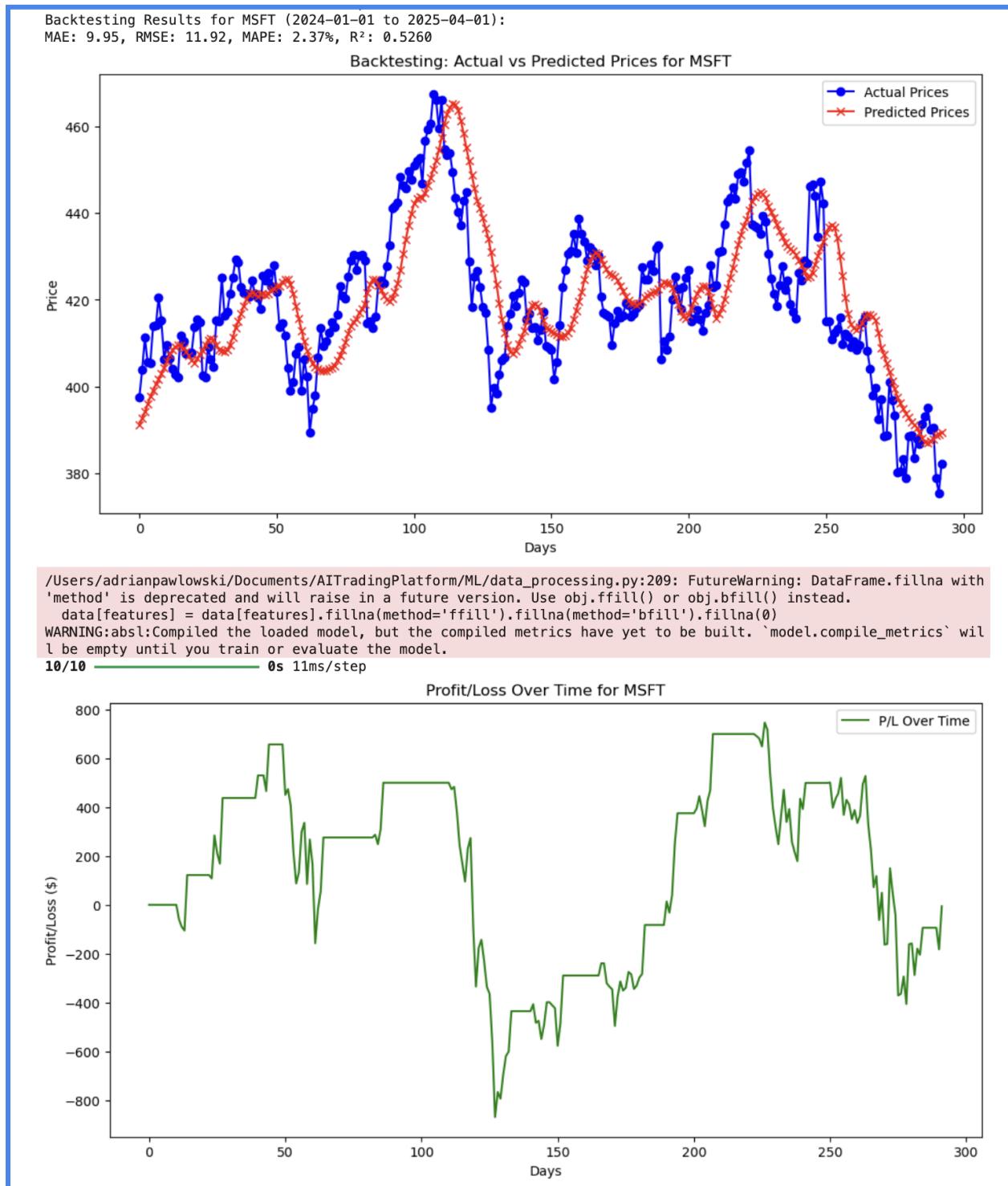


```
/Users/adrianpawlowski/Documents/AITradingPlatform/ML/data_processing.py:209: FutureWarning: DataFrame.fillna with
'method' is deprecated and will raise in a future version. Use obj.ffill() or obj.bfill() instead.
    data[features] = data[features].fillna(method='ffill').fillna(method='bfill').fillna(0)
WARNING:absl:Compiled the loaded model, but the compiled metrics have yet to be built. `model.compile_metrics` wil
l be empty until you train or evaluate the model.
10/10 ━━━━━━ 0s 11ms/step
```

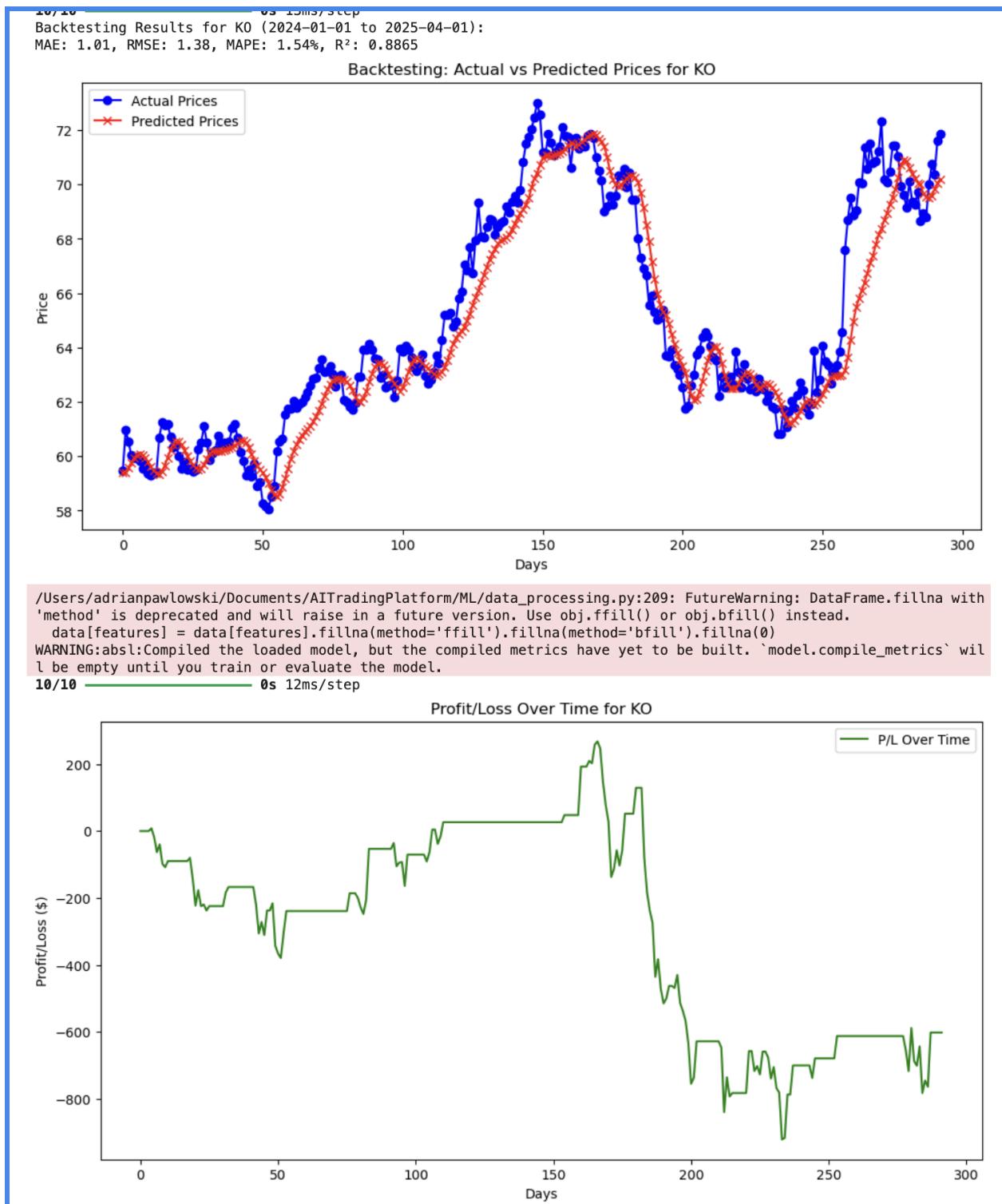
Profit/Loss Over Time for TSLA



Microsoft (MSFT)



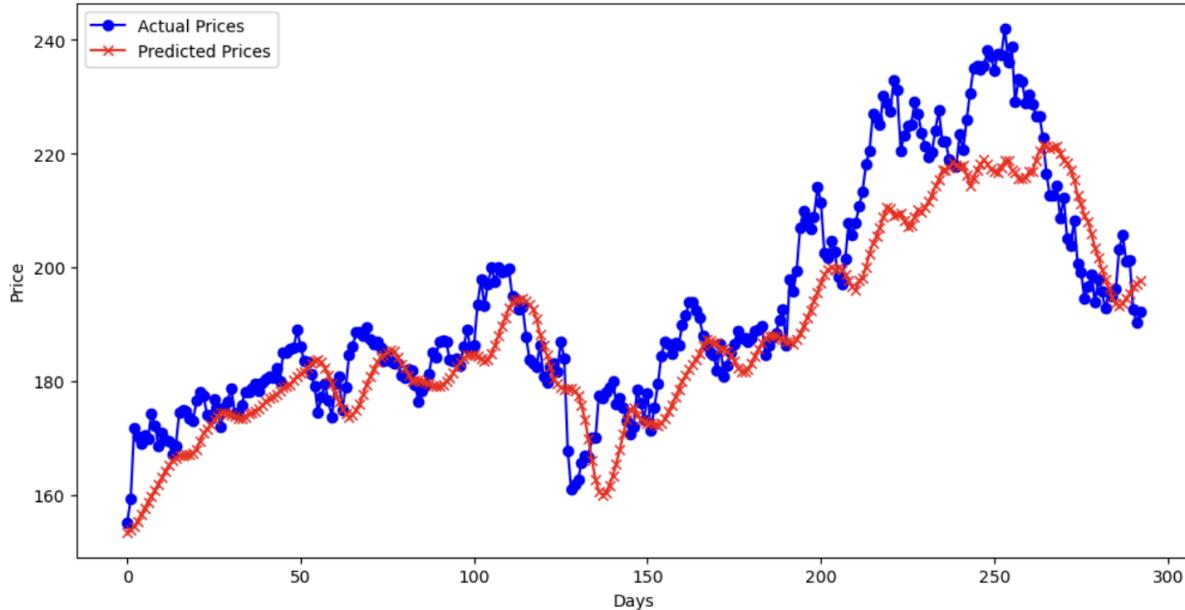
The Coca-Cola Company (KO)



Amazon (AMZN)

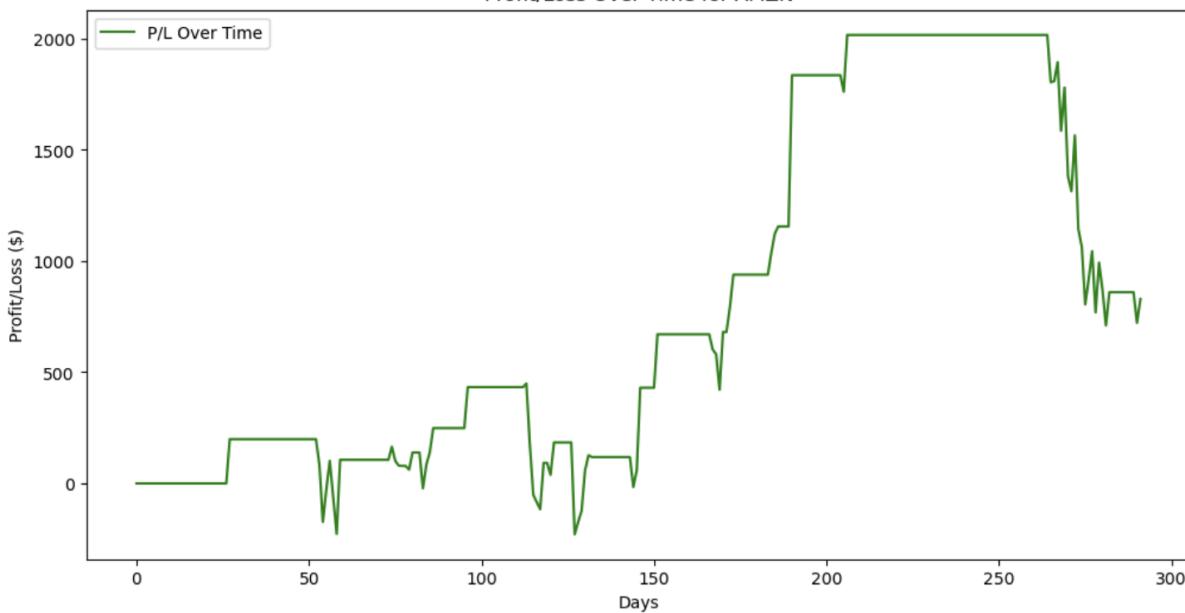
Backtesting Results for AMZN (2024-01-01 to 2025-04-01):
MAE: 7.80, RMSE: 9.78, MAPE: 3.92%, R²: 0.7484

Backtesting: Actual vs Predicted Prices for AMZN

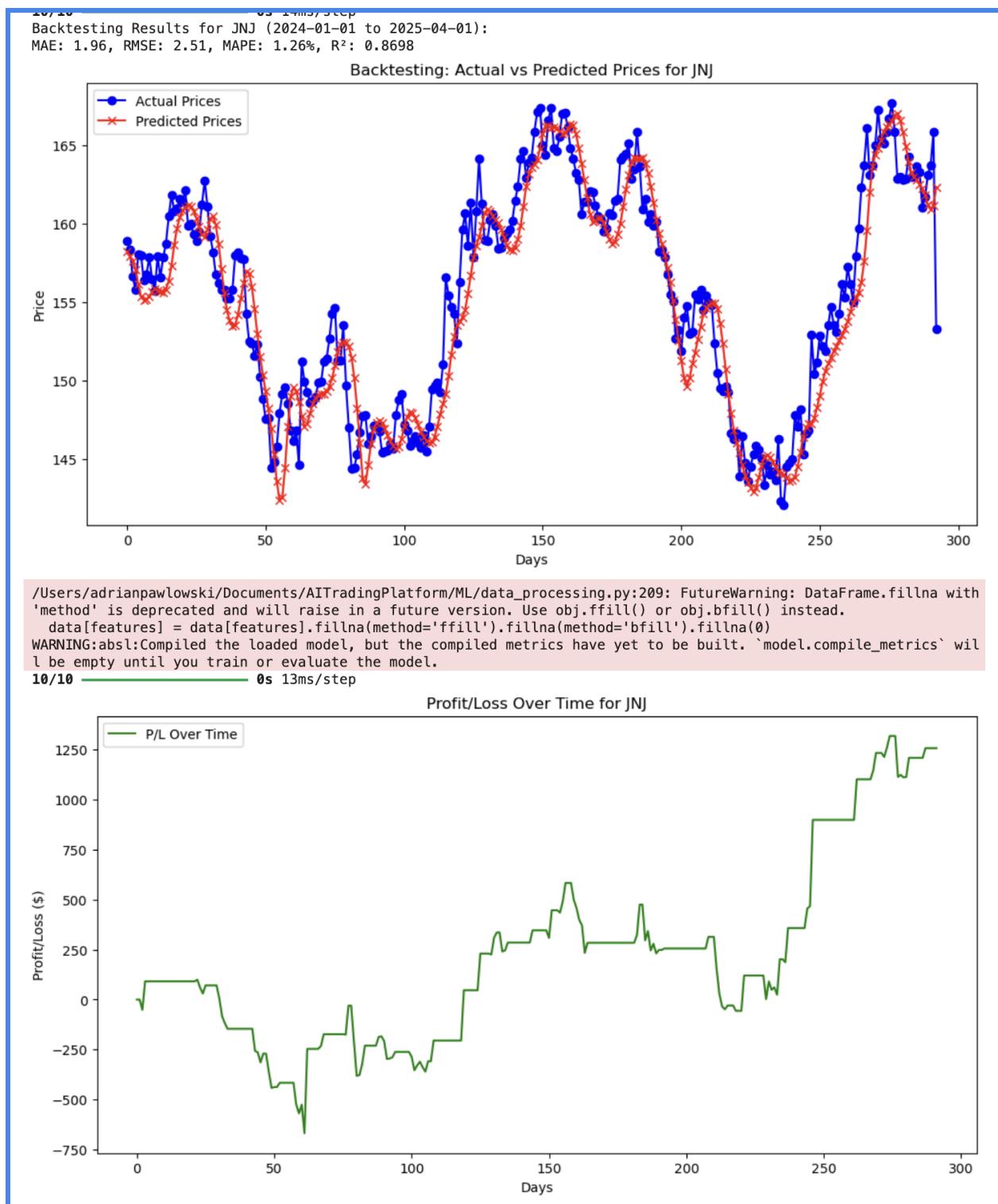


```
/Users/adrianpawlowski/Documents/AITradingPlatform/ML/data_processing.py:209: FutureWarning: DataFrame.fillna with  
'method' is deprecated and will raise in a future version. Use obj.ffmpeg() or obj.bfill() instead.  
    data[features] = data[features].fillna(method='ffill').fillna(method='bfill').fillna(0)  
WARNING:absl:Compiled the loaded model, but the compiled metrics have yet to be built. `model.compile_metrics` wil  
l be empty until you train or evaluate the model.  
10/10 ━━━━━━ 0s 13ms/step
```

Profit/Loss Over Time for AMZN



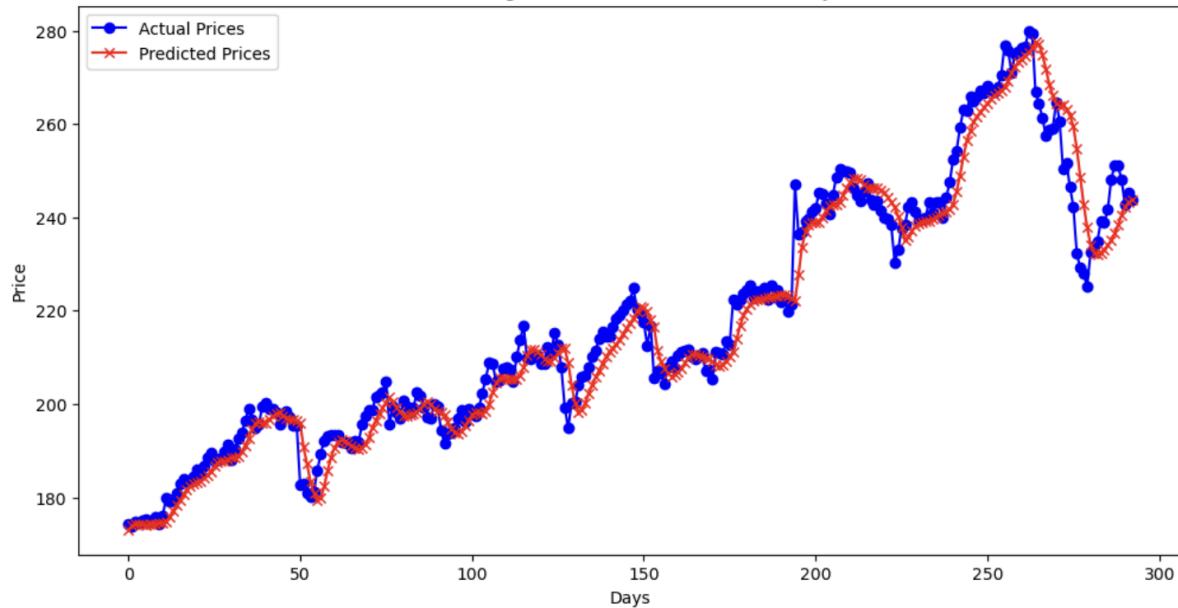
Johnson & Johnson (JNJ)



JP Morgan (JPM)

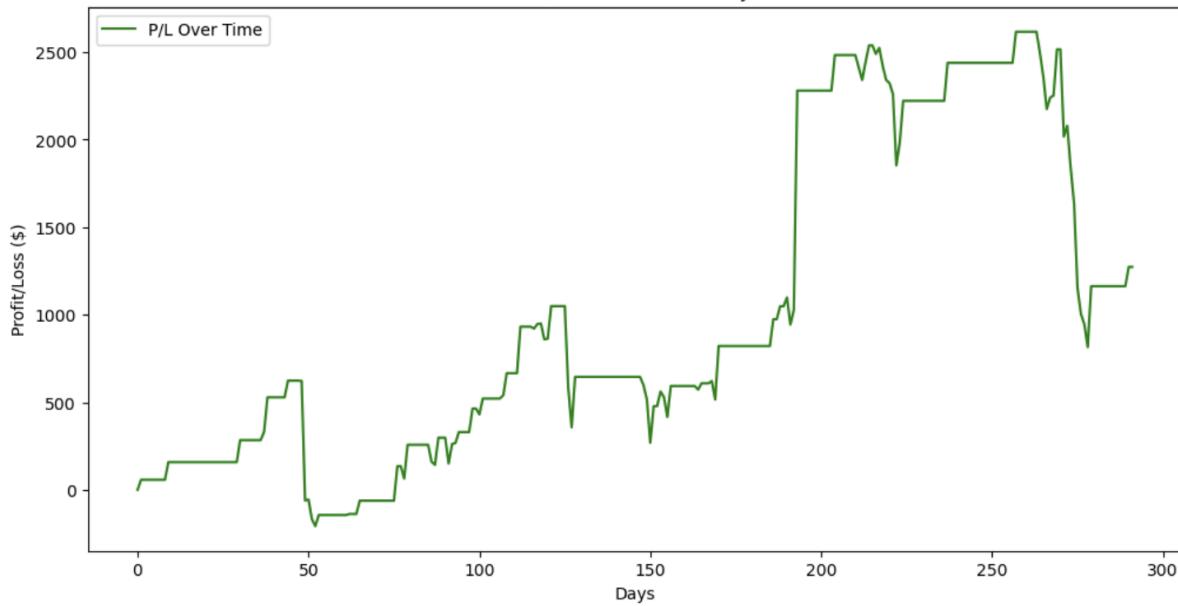
Backtesting Results for JPM (2024-01-01 to 2025-04-01):
MAE: 4.06, RMSE: 5.61, MAPE: 1.83%, R²: 0.9554

Backtesting: Actual vs Predicted Prices for JPM

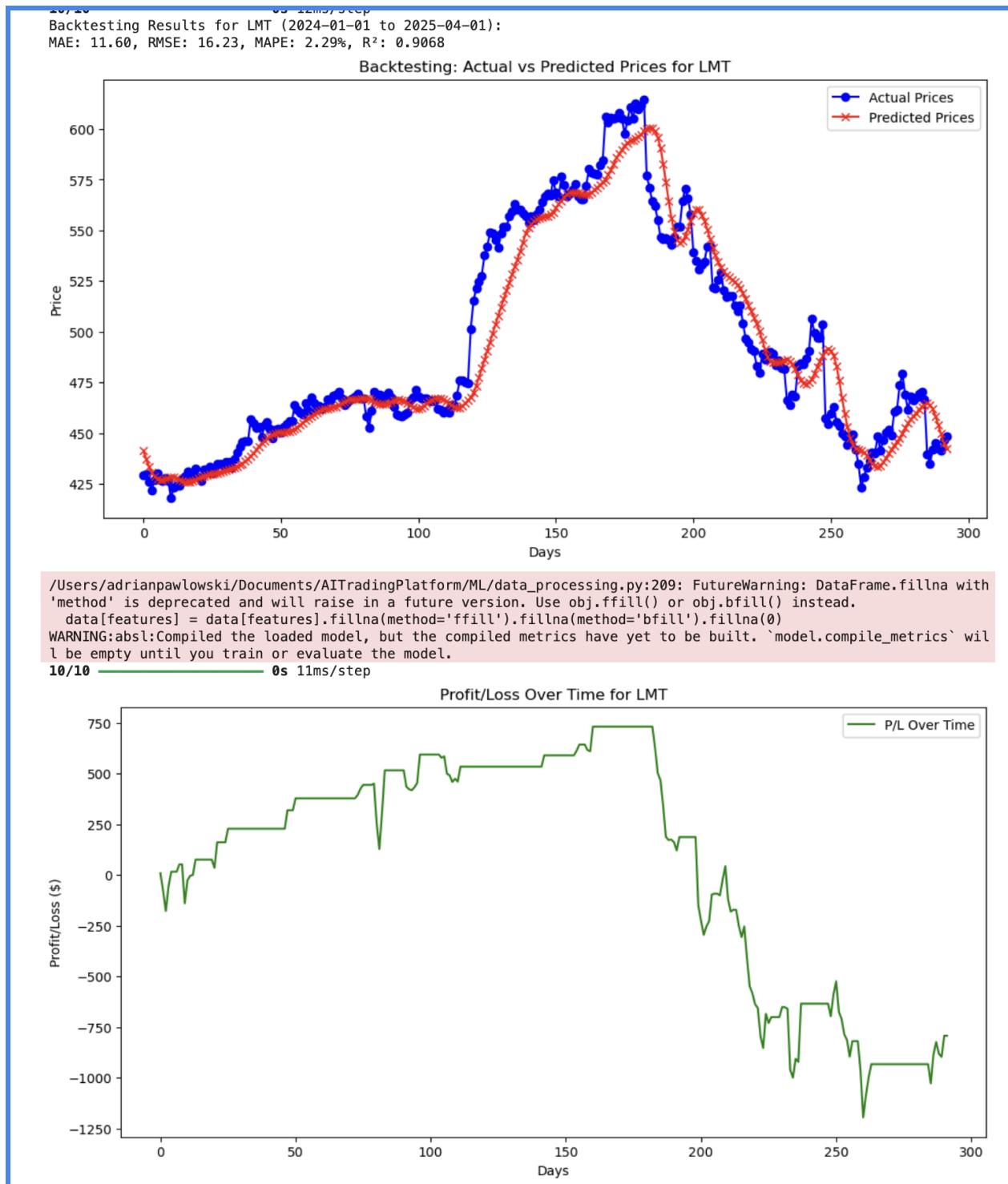


```
/Users/adrianpawlowski/Documents/AITradingPlatform/ML/data_processing.py:209: FutureWarning: DataFrame.fillna with  
'method' is deprecated and will raise in a future version. Use obj.ffill() or obj.bfill() instead.  
    data[features] = data[features].fillna(method='ffill').fillna(method='bfill').fillna(0)  
WARNING:absl:Compiled the loaded model, but the compiled metrics have yet to be built. `model.compile_metrics` wil  
l be empty until you train or evaluate the model.  
10/10 ━━━━━━ 0s 12ms/step
```

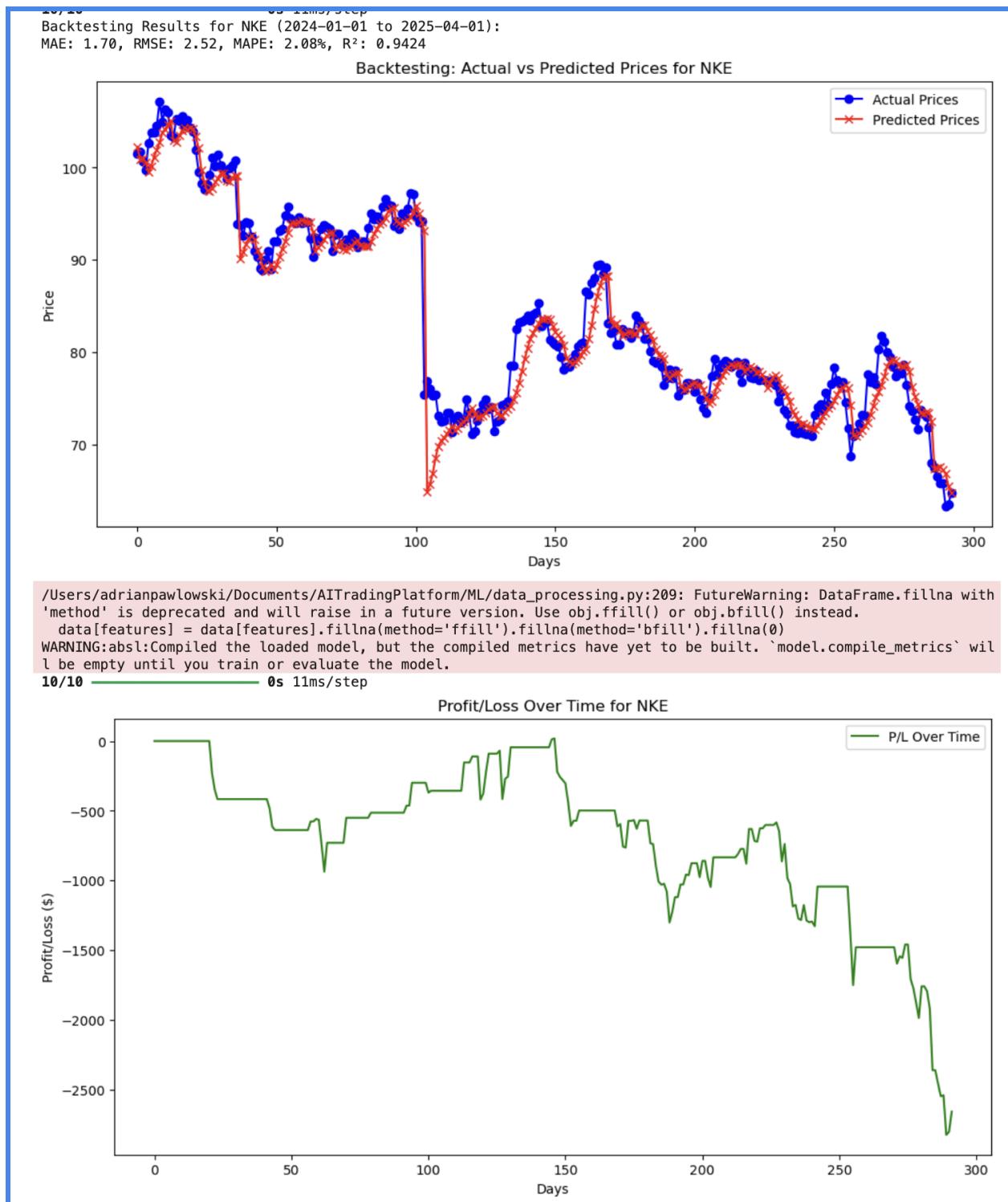
Profit/Loss Over Time for JPM



Lockheed Martin Corp (LMT)



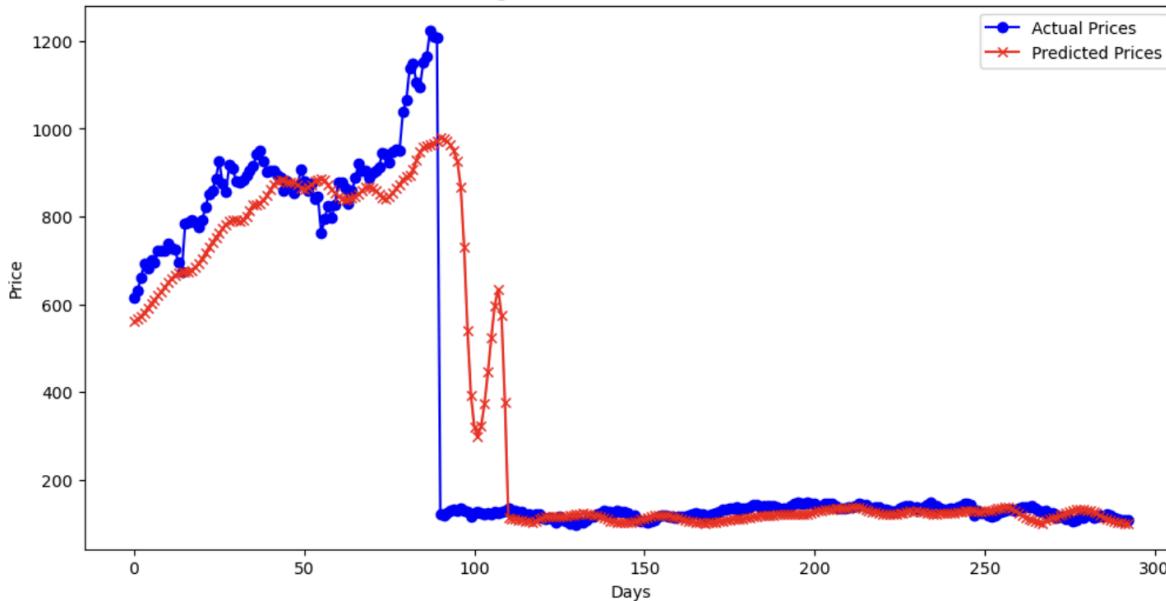
Nike Inc (NKE)



NVIDIA (NVDA)

Backtesting Results for NVDA (2024-01-01 to 2025-04-01):
MAE: 69.32, RMSE: 160.17, MAPE: 36.81%, R²: 0.7946

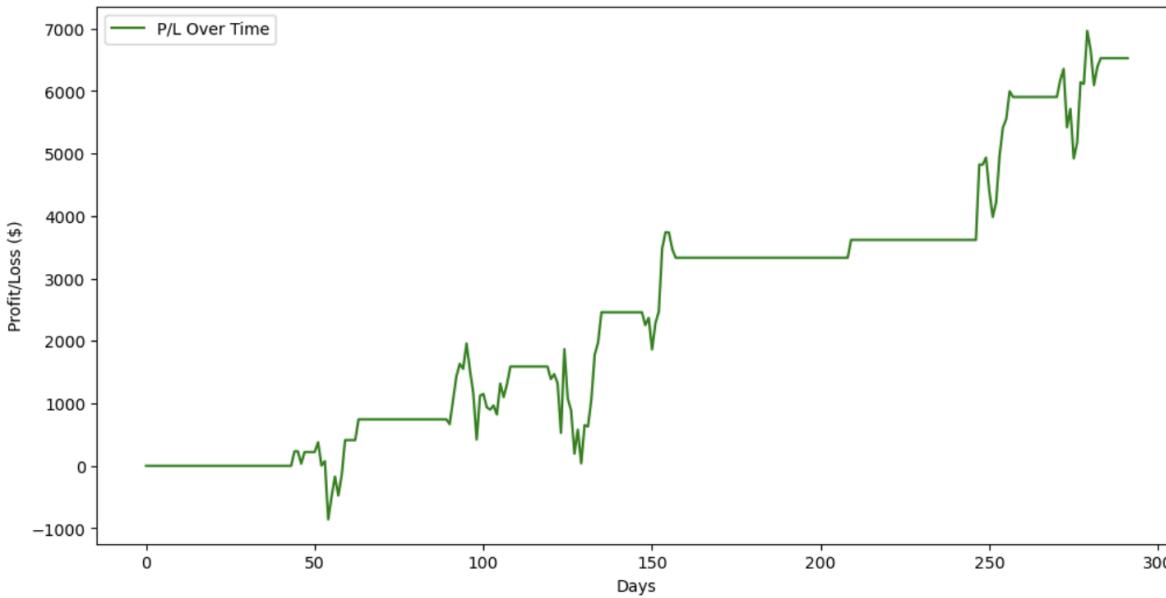
Backtesting: Actual vs Predicted Prices for NVDA



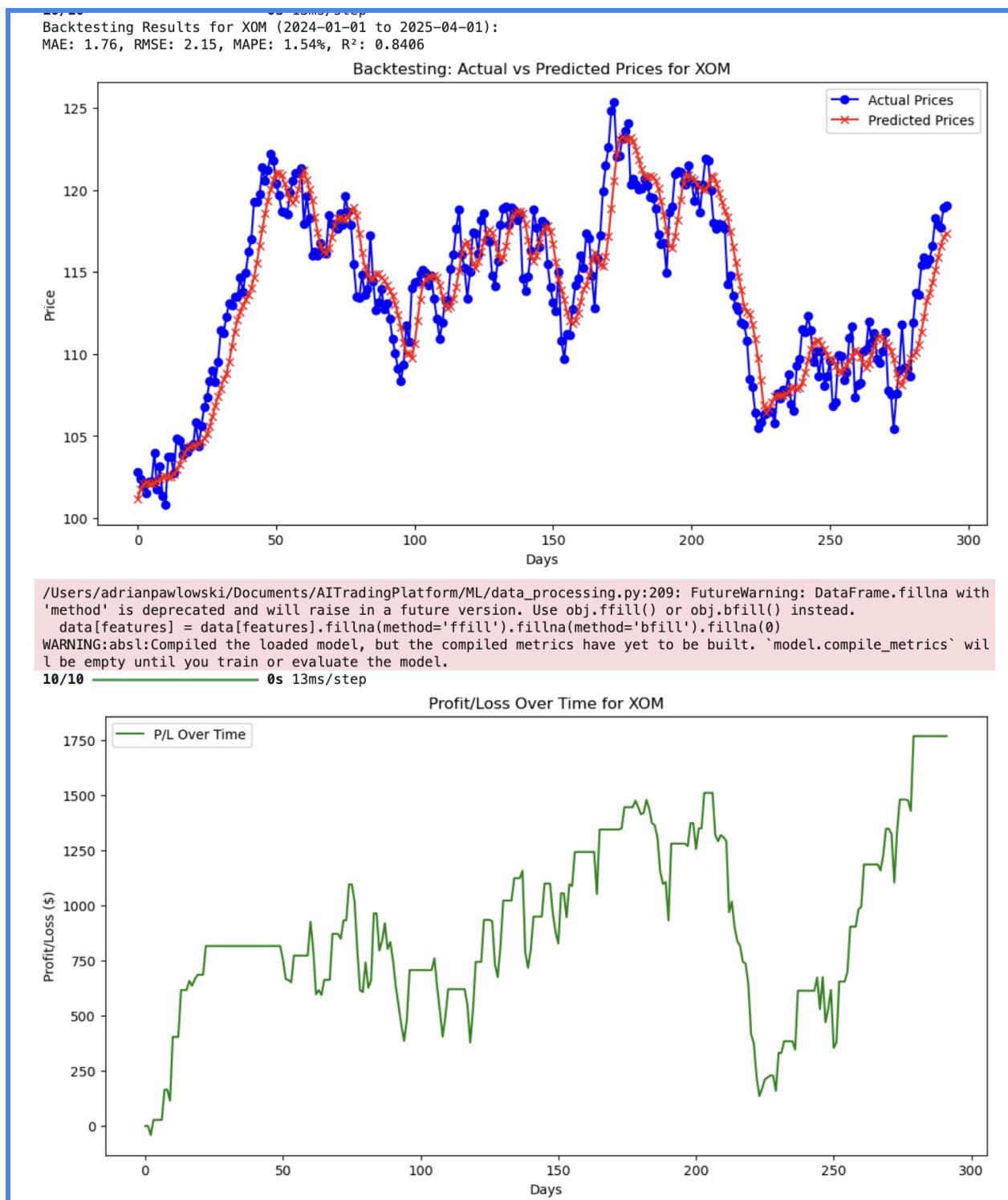
```
/Users/adrianpawlowski/Documents/AITradingPlatform/ML/data_processing.py:209: FutureWarning: DataFrame.fillna with
'method' is deprecated and will raise in a future version. Use obj.ffill() or obj.bfill() instead.
    data[features] = data[features].fillna(method='ffill').fillna(method='bfill').fillna(0)
WARNING:absl:Compiled the loaded model, but the compiled metrics have yet to be built. `model.compile_metrics` wil
l be empty until you train or evaluate the model.
```

10/10 0s 15ms/step

Profit/Loss Over Time for NVDA



Exxon Mobil (XOM)



4.6 LSTM Aggregate and Individual Results

```
==== Per-Ticker Portfolio Summary ====
  Ticker  Total P/L ($)  Overall % Gain  Annual % Gain  Sharpe Ratio \
0   TSLA      -1098.78      -10.9878      -8.901811      0.369395
1   MSFT        -5.66      -0.0566      -0.045338      0.409422
2    KO       -601.49      -6.0149      -4.847409     -0.943835
3   AMZN       829.34       8.2934       6.589822      1.013538
4   JNJ        1257.80      12.5780      9.954650      0.463587
5   JPM        1273.46      12.7346     10.077144      1.113152
6   LMT       -792.06      -7.9206      -6.395955      0.113542
7   NKE       -2657.74      -26.5774     -21.921511     -1.265636
8   NVDA       6525.62      65.2562      49.534532      1.214809
9   XOM        1766.62      17.6662      13.917673      2.263658

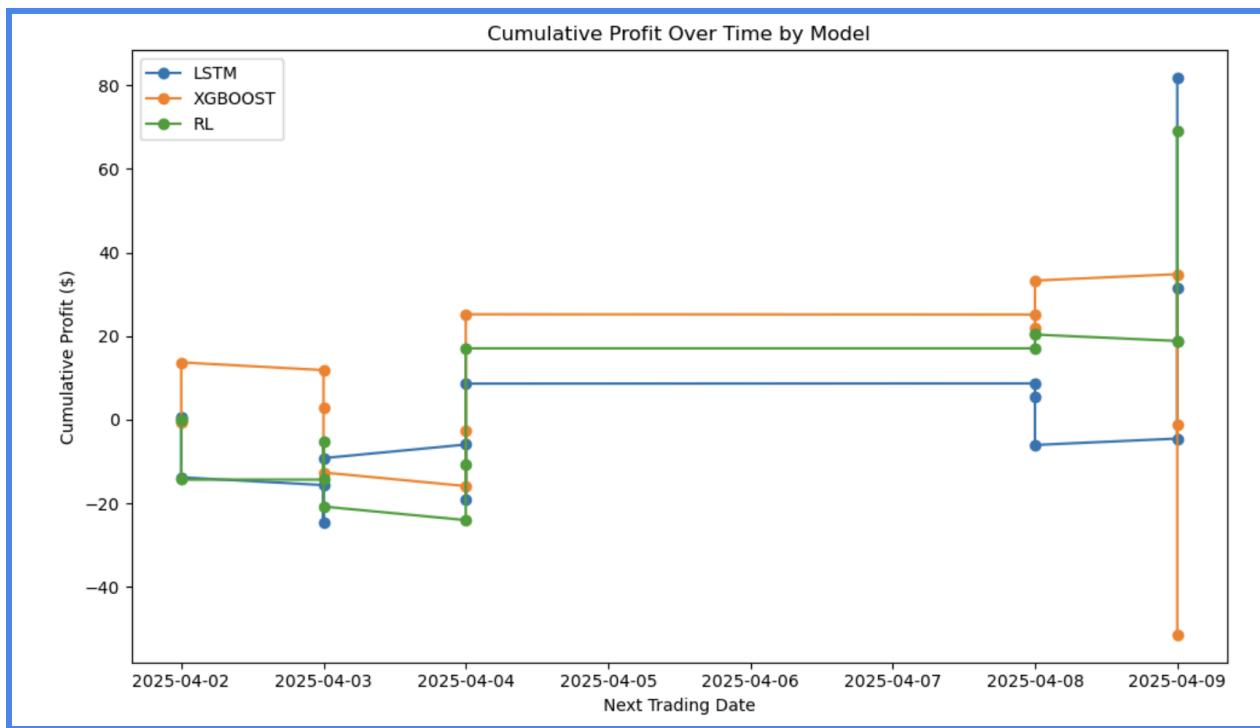
  Max Drawdown ($)
0            -1630.92
1            -869.43
2            -920.91
3            -228.71
4            -668.35
5            -207.55
6           -1194.26
7           -2823.85
8            -857.12
9            -41.16

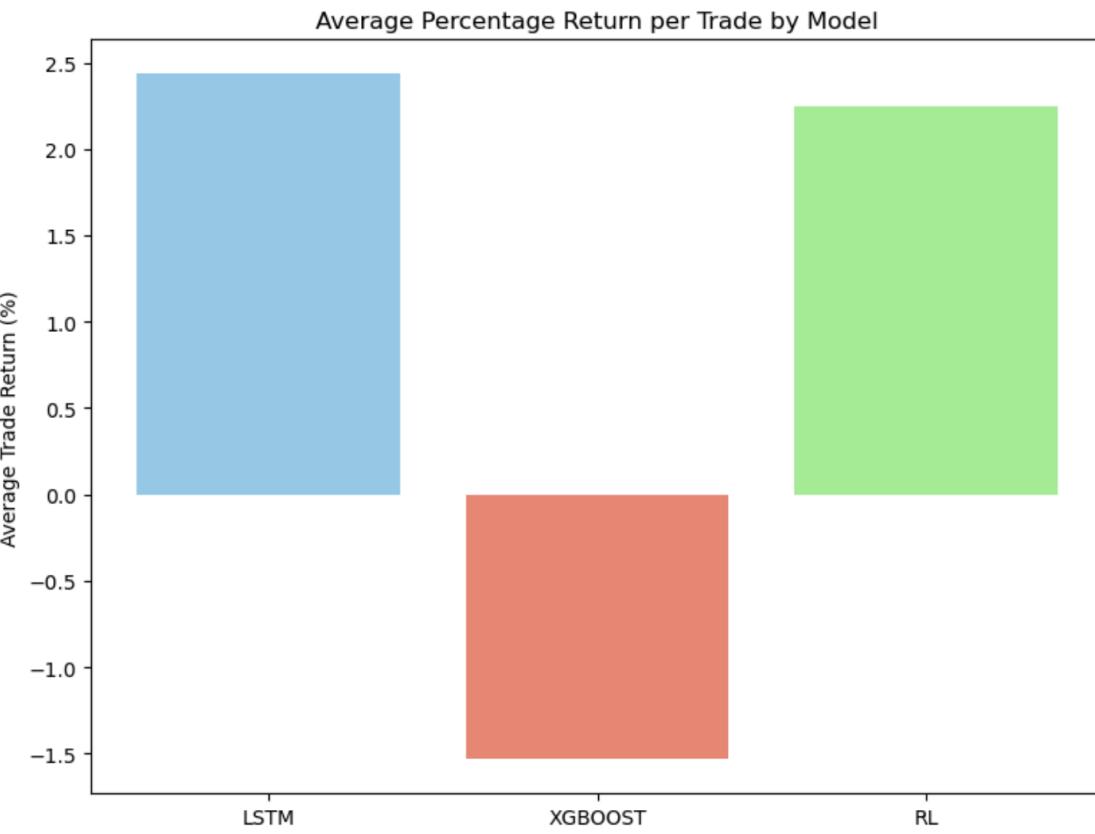
==== Aggregated Portfolio Summary (across all tickers) ====
Aggregated Total P/L ($): 6497.11
Aggregated Overall % Gain: 6.50
Aggregated Annual % Gain: 2.84
Aggregated Sharpe Ratio: 1.39
```

4.6 Live Trading Results

Metrics Evaluated:

- Cumulative Profit Over Time per Model
- Average Percentage Return per Trade by Model
- Cumulative Percentage Returns
- Annualized Returns





5. Discussion and Analysis

5.1 Reinforcement Learning Model

The **Reinforcement Learning** model met key financial thresholds, including final balance and realized P/L, across a diverse set of stocks. It performed well with volatile stocks like **TSLA** and **NVDA** while maintaining stability with lower-risk names like **KO** and **JNJ**. The model's reward structure allowed for effective decision-making over time, balancing short-term actions with long-term returns. Overall, the RL agent demonstrated adaptability and consistent portfolio growth.

5.2 XGBoost Model

XGBoost delivered solid results, especially on high-volume stocks like **MSFT** and **AMZN**, where price movements were more pattern-driven. While a few individual stocks underperformed, the overall portfolio met the required profit and gain criteria. Its speed and feature transparency make it a strong candidate for real-time signal generation. It serves as a lightweight but valuable layer in our overall trading strategy.

5.3 LSTM Model

The **LSTM model** showed strength in predicting price trends for stable stocks like **KO** and **AMZN**, thanks to its ability to learn from sequential data. Error metrics and risk-adjusted performance (**Sharpe Ratio > 1.2**) were strong at the portfolio level. Some stocks underperformed individually, but overall gain and annualized return comfortably surpassed expectations. The model proves effective for medium- to long-term forecasting within our platform.

5.4 Live Trading

Our live trading tests showed **profitable performance overall**, with both the LSTM and RL models generating positive returns. The **LSTM model** earned **\$81.74** in profit with an average return of **2.44%** and a **Sharpe Ratio of 0.29**. The Reinforcement Learning model returned **\$69.18** with a **2.25%** average return and a **Sharpe Ratio of 0.35**, reflecting strong risk-adjusted gains. While the XGBoost model underperformed with a loss of **-\$51.44** and a negative return, the total combined profit across models was **\$99.48**, validating our platform's effectiveness in live conditions and confirming system reliability and real-time prediction accuracy.

```
--- Overall Model Performance Metrics (Dollar) ---
LSTM MAE: 13.6935, RMSE: 18.2848, Directional Accuracy: 60.00%
XGBOOST Directional Accuracy: 53.33%
RL Directional Accuracy (excluding Hold): 40.00%
NLP Sentiment / Return Correlation: 0.0157

--- Overall Portfolio Simulation per Model (Dollar Profits) ---
LSTM: {'Total Profit ($)': 81.73999999999994, 'Average Profit ($)': 5.44933333333329, 'Sharpe Ratio': 0.287914717
73189414, 'Number of Trades': 15}
XGBOOST: {'Total Profit ($)': -51.44000000000001, 'Average Profit ($)': -3.42933333333334, 'Sharpe Ratio': -0.176
51584538537946, 'Number of Trades': 15}
RL: {'Total Profit ($)': 69.17999999999992, 'Average Profit ($)': 6.91799999999992, 'Sharpe Ratio': 0.34799911246
42655, 'Number of Trades': 10}
```

```
--- Overall Portfolio Average Trade Percentage Returns ---
LSTM Average % Return: 2.44%
XGBOOST Average % Return: -1.53%
RL Average % Return (excluding Hold): 2.25%

--- Overall Cumulative Percentage Returns ---
LSTM Cumulative % Return: 2.35%
XGBOOST Cumulative % Return: -1.48%
RL Cumulative % Return: 2.58%

--- Overall Annualized Returns ---
LSTM Annualized Return (%): 236.21%
XGBOOST Annualized Return (%): -54.06%
RL Annualized Return (%): 278.54%

--- Overall Combined Profit across Models (Dollar) ---
{'LSTM ($)': 81.73999999999994, 'XGBOOST ($)': -51.44000000000001, 'RL ($)': 69.1799999999992}
```