MSAI 451 – Programming Assignment 1

Author: Divyanka Thakur

---

1. The Challenge: Building a Predictive Edge

The core objective of this project was to determine if an **active-management style classifier** could reliably predict the **next-day price direction (up or down)** for a high-growth asset, **NVIDIA Corporation (NVDA)**, using only historical price and volume data.

In the spirit of financial forecasting (Information → Forecast → Action), we tested whether carefully engineered technical features—lags, spreads, and exponential moving averages (EMAs)—contain exploitable, forward-looking information.

Asset & Target

- **Asset Chosen:** NVIDIA Corporation (NVDA) daily OHLCV data.
- **Target (Targett):** A binary variable indicating a positive log-return (i.e., the price closed higher than the previous day).

Success Standard

Success was primarily defined by achieving **out-of-sample classification accuracy materially above 50%** (a coin-flip baseline) and demonstrating this stability across cross-validation folds. Secondary metrics included **F1 Score** and **ROC-AUC**.

---

2. Data Strategy & Feature Engineering ⚙️

The modeling pipeline was built on a foundation of clean data and mathematically sound feature engineering, leveraging the speed of the **Polars** DataFrame library.

Data Source & Preparation

- **Source:** Local CSV file (`msds_getdata_yfinance_nvdl.csv`).
- **Cleaning:** Dates were parsed and sorted, and non-essential columns (Dividends, Stock Splits) were dropped. Initial rows with nulls, created during the lag and EMA calculations, were removed to maintain data integrity.

Creating the Features

To prevent **information leakage**, all features were based on data available *before* the prediction date.

Feature Category	Description	Examples
Lags	Previous day's prices and volume.	CloseLag1..3, VolumeLag1..3
Candlestick Spreads	Measures of volatility and range *within* a day.	HML (High − Low), OMC (Open − Close)
Momentum/Smoothing	Exponentially weighted moving averages (EMAs) to capture short-term trends.	CloseEMA2, CloseEMA4, CloseEMA8 (computed off lagged close)

The final modeling table deliberately **excluded contemporaneous prices** (Close, Open, High, Low, Volume) to ensure a true time-series prediction setup.

Initial Diagnostics

- **Class Balance:** The up-day class (Target=1) was slightly dominant at **55.2%** (383/694), indicating a modest inherent bullish drift in the sample period.
- **Feature Correlation:** A heatmap (Figure 3) confirmed high, expected correlations among closely related features, such as adjacent price lags and EMAs. The choice of **XGBoost** as a model family handles this multicollinearity implicitly.

---

3. Research Design & Model Selection

To rigorously test for a genuine predictive edge, we employed **time-aware cross-validation** and a systematic hyperparameter search.

Cross-Validation (CV) Methodology

- **Technique: TimeSeriesSplit** (5 folds, with a gap of 10 days) was used to ensure that the model was only trained on data chronologically *before* the test data, mimicking a real-world deployment.

- **Baseline CV Results:**
  - **Average CV Accuracy: 0.503**
  - **Std of CV Accuracy: 0.035**

This baseline result is critically important: **the average signal on daily direction is extremely weak**, lending support to the **Efficient Market Hypothesis (EMH)** which suggests that daily price movements are essentially random and unpredictable.

Model & Hyperparameter Tuning

- **Model Family: XGBoost Classifier** (objective=`binary:logistic`) was selected for its robust performance in classification tasks and its ability to capture non-linear relationships.
- **Tuning: RandomizedSearchCV** optimized the model based on CV accuracy, searching over common hyperparameters (max_depth, learning_rate, n_estimators, etc.).

Best CV Model Parameters	Result
max_depth	5
n_estimators	788
learning_rate	≈0.0573
Best CV Accuracy (in a single fold)	**0.541**

The best-performing fold achieved a **modest 54.1% accuracy**, suggesting an **occasional, small edge**. However, the overall fold-average remained near 50%, highlighting the challenge.

---

4. Final Results & Interpretation

4.1 Out-of-Sample Performance: The True Indicator

The CV metrics are the **reliable measure of generalization** to unseen data.

- **Mean Out-of-Sample Accuracy: 0.503**
- **Best-Case Out-of-Sample Accuracy (single fold): 0.541**

Takeaway: The evidence for a stable, exploitable trading edge at the daily frequency using only these features is **not conclusive**. The model struggles to consistently beat chance, aligning with the expected difficulty of short-term market forecasting.

4.2 Final Fit Diagnostics: The Caveat

After tuning, the model was fit on the *entire* historical dataset to provide a comprehensive look at its predictive capacity. These metrics are **in-sample** and therefore **optimistic**.

- **Resubstitution (In-Sample) Accuracy: 0.818**
- **ROC-AUC: ≈0.81** (Figure 1)
- **Confusion Matrix (Figure 2):**
    - True Positives (TP)=336
    - True Negatives (TN)=232

These highly favorable in-sample results confirm the model can **fit complex patterns** within the full data, but the stark difference between the 81.8% in-sample accuracy and the 50.3% out-of-sample accuracy underscores the severity of **overfitting** when modeling daily stock direction.

---

5. Conclusions & Future Direction

Conclusion

Classification of daily price direction for NVDA using only price/volume-derived technical features is **extremely challenging**. The average out-of-sample accuracy of 0.503 suggests that the daily movements are dominated by noise, consistent with established financial theory.

Next Steps: Seeking a Stable Edge

To move from a weak signal toward a potentially profitable trading system, future work should focus on reducing noise and incorporating external market context:

1. **Adjust the Time Horizon:** Shift the target to a **longer-term label** (e.g., predicting direction at t+5 days) to potentially capture more signal and less day-to-day noise.
2. **Incorporate Market Context:** Introduce **exogenous features** like the S&P 500 return (SPY) or VIX volatility levels. Asset-specific features may only capture α, but market-wide features capture β (systematic risk).

3. **Enhance Robustness:** Implement **walk-forward evaluation** with a rolling window to better simulate a real trading deployment, and use metrics like **Balanced Accuracy** or **MCC** which are less susceptible to mild class imbalance.

---

6. Deliverables

The complete project, including code and data, is available in the designated public GitHub repository.

- `451_pa1_jump_start_v001.py` (End-to-end pipeline)
- `getdata_yfinance.py` (Extracting csv file)
- `msds_getdata_yfinance_nvdl.csv` (Input data)
- `Figure_1.png` (ROC Curve)
- `Figure_2.png` (Confusion Matrix)
- `Figure_3.png` (Correlation Heatmap)