project wrap-up deck ↘

NVIDIA | Predicting Tomorrow's Price Direction from Market Signals

PRESENTED BY:
Fabien Sportouch, Guillaume Rougier, Allan Vu, Julie Wallet

content ↘

price predictor

what

We built a machine learning pipeline to predict NVIDIA stock price movement (up or down) for the next day, using historical stock data and technical indicators.

why

In volatile markets, anticipating short-term price direction is key for active investors. Traditional tools lack predictive power or are overly reactive. Our project leverages data science to offer a smarter decision support tool.

how

- Smart Features: We engineered technical indicators (RSI, MACD, Bollinger Bands...) from historical data to capture real trading signals.
- Multiple Models: We trained and compared 4 classifiers (Logistic Regression, Decision Tree, SVC, Random Forest) to find the best fit for financial time series.
- Visual Evaluation: Our models were tested on real timelines with error tagging to ensure interpretability and relevance.

project pipeline ↘

01

02

03

Problem Framing

- Objective: Predict if NVIDIA's stock will go up or down the next day
- Reframed as a binary classification problem
- Focused on actionable signals for traders or portfolio managers

Feature Engineering

- Extracted technical indicators: RSI, MACD, CCI, Bollinger Bands
- Integrated external signals: BTC price returns
- Created lag features to capture short-term dynamics

Modeling & Evaluation

- Benchmarked Logistic Regression, Decision Tree, SVM, and Random Forest
- Assessed accuracy and stability on test set
- Visual inspection via prediction timeline and confusion matrix helped reveal behavioral patterns

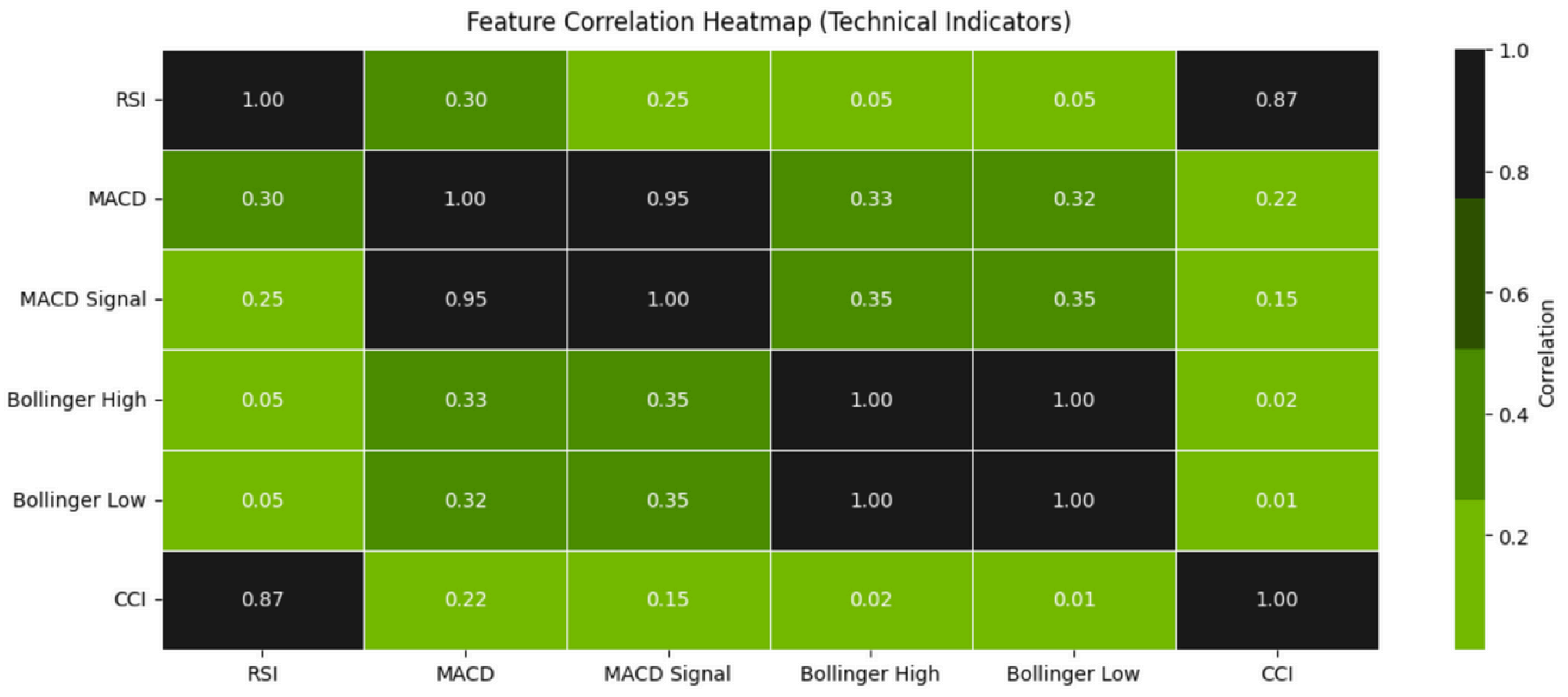
technical indicators

- RSI – momentum & overbought/oversold zones
- MACD & Signal – trend strength & direction
- Bollinger Bands – volatility via price deviation
- CCI – cyclical trend detection

external returns (daily %)

- Bitcoin & Ethereum – crypto sentiment
- QQQ, S&P 500 – market direction
- VIX, Oil – macro volatility & sentiment
- AMD – peer semiconductor performance

features market signals



target definition ↘

```
df["Target"] = (df["Close"].shift(-1) > df["Close"]).astype(int)
```

how we built it

- 1 → Tomorrow's price goes up
- 0 → Price stays flat or drops

why it matters

- Reflects real trading logic: buy or hold
- Avoids regression on tiny fluctuations
- ~50/50 class distribution = no class imbalance
- Focuses on trend direction, not prediction amplitude

model strategy ↘

Model Lineup

- Logistic Regression – Linear baseline
- Decision Tree – Rule-based learner
- SVC – Non-linear margin classifier
- Random Forest – Ensemble benchmark
- All models trained on same binary target

Preprocessing

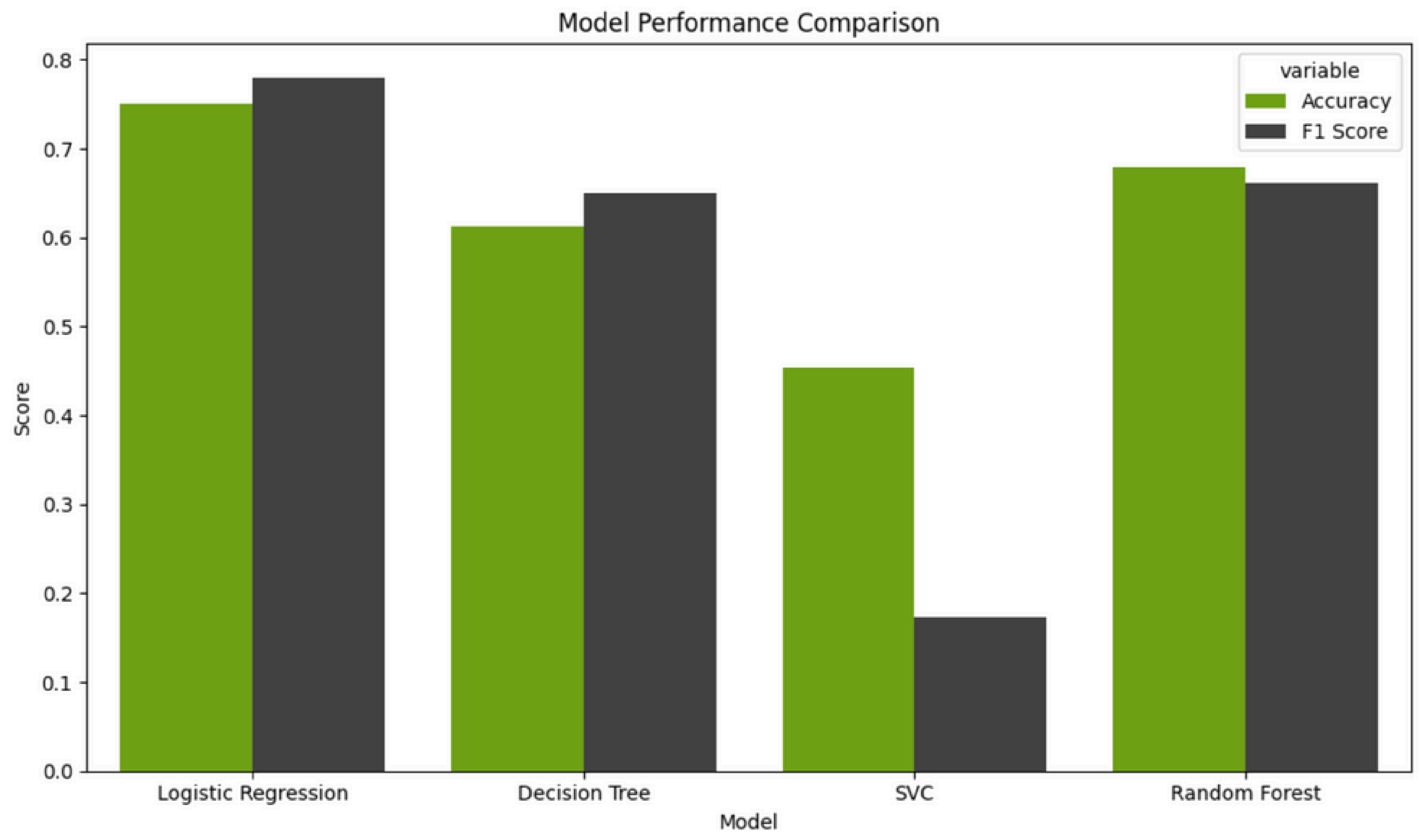
- Train/test split – Randomized to avoid lookahead bias
- Scaling – Applied where needed (e.g., SVC, Logistic)
- Ensured consistent input space across models

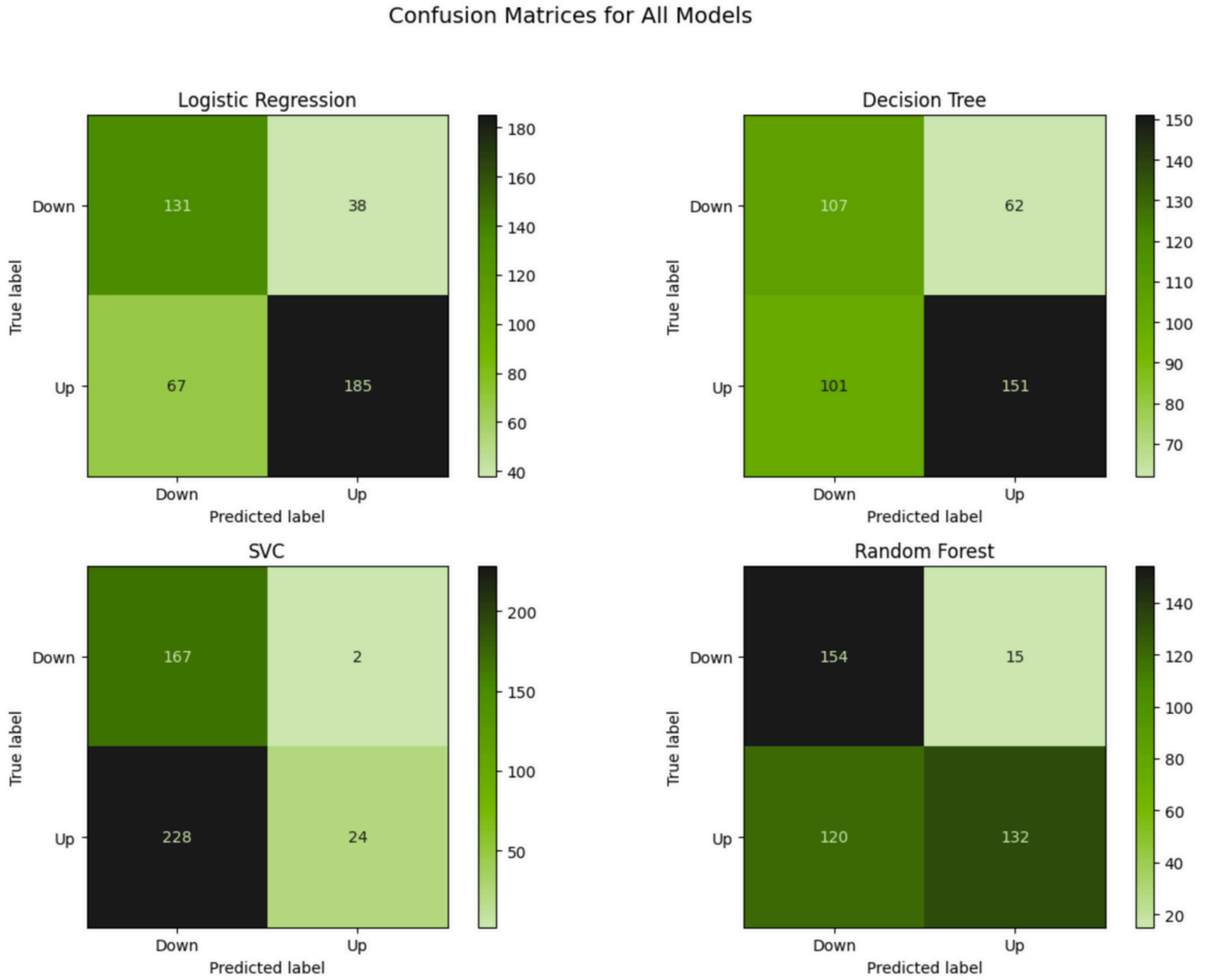
Evaluation Metrics

- Accuracy – % of correct up/down predictions
- F1 Score – Balances precision & recall
- Registered models to jrjModelRegistry for reproducibility and future use

model results ↘

	Accuracy	F1 Score
Logistic Regression	0.75	0.78
Decision Tree	0.61	0.65
SVC	0.45	0.17
Random Forest	0.68	0.66





30-DAY TEST SEGMENT

- Visual display of actual NVIDIA stock prices over a 30-day test period, with each model's predictions plotted on top.

VISUAL ENCODING

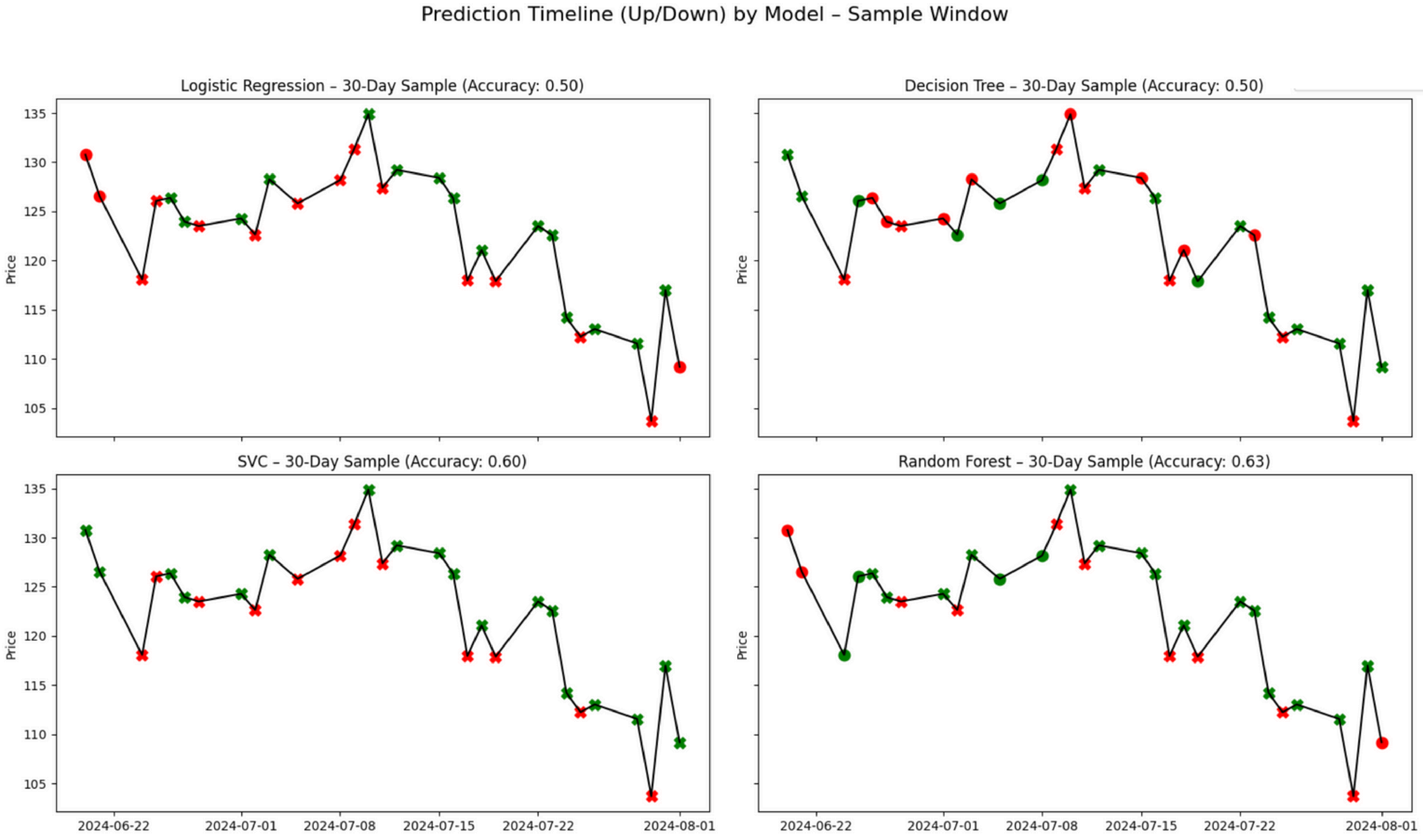
- Green markers = correct predictions
- Red markers = incorrect predictions
- ● / ✚ Different markers for “Up” and “Down” classes

WHY IT MATTERS

- Reveals real model behavior on unseen data
- Helps spot weaknesses, e.g., overfitting or missed reversals
- Builds trust through visual inspection
- Shows where and when models fail or succeed — not just how often

INSIGHT

- this view helps assess models beyond scores, revealing patterns in their mistakes.
- random forest stands out with fewer directional errors during high-volatility periods.



real predictions

model insights ↘ takeways

BEST FIGHT FOR DEPLOYMENT

Random Forest stands out for its strong performance and ability to generalize well on unseen data.
It offers the highest accuracy and a reliable F1-score, making it the most effective at predicting directional movements.
Logistic Regression remains a viable production-ready option — offering speed, interpretability, and robust baseline performance.

Model	Strengths	Weaknesses
Logistic Regression	Interpretable, robust baseline	Misses nonlinear patterns
Decision Tree	Captures decision rules, fast	Overfits on noise and small fluctuations
SVC	Sensitive to subtle shifts, margin-based logic	Very low F1, poor generalization, slower inference
Random Forest	Best balance of variance & bias, top F1 Score	Harder to interpret, may learn noise

model limits insights

	risk	mitigation	alternative
01	Overfitting to recent patterns	Used train-test split on unseen 30-day segment; added dropout where possible	Future: Time Series Cross-Validation for more robust estimation
02	Noisy or misleading market signals	Filtered highly correlated features (e.g., MACD Signal) via correlation heatmap	Add feature importance analysis or SHAP explanations
03	Lack of interpretability for trading decisions	Kept Logistic Regression as a transparent baseline	Integrate LIME/SHAP visual explanations to complex models

Even the best-performing model (Logistic Regression) reaches only 75% accuracy.
This is acceptable for short-term directional trading, but further improvements are needed to handle volatility and explain predictions in high-stakes use cases.

next steps

Deploy in simulated trading

- Test real-time performance in market-like environments

Add new features

- Integrate volume, news sentiment, macro indicators

Cross-stock testing

- Apply pipeline to AMD, Intel, S&P500

Build dashboard

- Create an interface for exploration and demo use

next steps ↗ ethics

ethical considerations

No financial advice

- Predictions are directional signals, not trading recommendations

Transparent logic

- Avoid black-box models in high-stakes financial contexts

Bias control

- Regular checks to prevent overfitting or noise exploitation

Human-in-the-loop

- Always combine predictions with expert judgment

thank you for following our journey

questions welcome ↘